AD-A096 597 BOEING MILITARY AIRPLANE CO SEATTLE WA F/6 1/3
ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM. VOLUME --ETC(U)
SEP 80 C L WEST, B R UMMEL, R F YURCZYK F33615-79-C-3407
UNCLASSIFIED AFWAL-TR-80-3014-VOL-1 NL

1 OF 3

AD A
096597

AFWAL-TR-80-3014
VOLUME I

ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM

Christopher L. West
Brian R. Ummel
Roger F. Yurczyk

Boeing Military Airplane Company
P.O. Box 3707
Seattle, Washington 98124

II

2

SEPTEMBER 1980

Final Report for Period May 1979 to September 1980

Flight Dynamics Laboratory
Air Force Wright Aeronautical Laboratory
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

Lanny A. Jines

LANNY A. JINES
Project Engineer

James M. Peters

JAMES M. PETERS, Acting Chief
Aircrew Escape Group
Crew Escape & Subsystems Branch
Vehicle Equipment Division

FOR THE COMMANDER

Ambrose B. Nutt

AMBROSE B. NUTT
Director
Vehicle Equipment Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/FIER, Wright-Patterson AFB, OH 45433 to help us maintain a current mailing list."

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/17 February 1981 — 75

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

19 18 6 10 15 16 17 11 12 9

1. REPORT NUMBER
AFWAL-TR-80-3014 - ~~Volume I~~ VOL-I

2. GOVT ACCESSION NO.
AD-A096 597

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM.
VOLUME I.

5. TYPE OF REPORT & PERIOD COVERED
Final rept.
May [redacted]79-[redacted] Sept[redacted]80

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Christopher L. West
Brian R. Ummel
Roger F. Yurczyk

8. CONTRACT OR GRANT NUMBER(s)
F33615-79-C-3407

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Boeing Military Airplane Company
P.O. Box 3999
Seattle, Washington 98124

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
24020328

11. CONTROLLING OFFICE NAME AND ADDRESS
Flight Dynamics Laboratory
Air Force Wright Aeronautical Laboratories
Air Force Systems Command, WPAFB, Ohio 45433

12. REPORT DATE
Sep[redacted]80

13. NUMBER OF PAGES
675

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
690

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
STATEMENT A - "Approved for public release; distribution unlimited."

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
VOLUME II containing the source code listing of the EASY5 computer program is proprietary data to Boeing Military Airplane Company for a three year period following the completion of the contract, unless restriction agreement renewed by Contractor and Controlling office.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Computer Simulation
Stability & Control
Ejection Seat

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
High performance combat aircraft have extended the maneuvering/operating range into regimes that exceed the capabilities of current ejection seat systems. One of the problems encountered involves the unstable rotational characteristics of the typical ejection seat, resulting in a decreased probability of survival due to the reorientation of the ejecting crewmember into an attitude less tolerant to acceleration. Furthermore, an unstable ejection seat may neither clear the airframe, nor provide adequate ground clearance. The capability to simulate the trajectory of an escape system, and to determine its stability

411 593

#20 continued.
characteristics using classical stability and control methods, is required to enhance the development of both active and passive stability augmentation systems.

The Simulation and Analysis of In-Flight Escape System Techniques (SAFEST) computer program, developed by the AFFDL for the analysis of occupied ejection seat stability characteristics, is a six degree of freedom simulation of an ejection system. SAFEST uses a fourth order Runge-Kutta integrator with a fixed time step to calculate the trajectories for the seat/man, man alone, airplane, drag parachute, and the recovery parachute. However, SAFEST does not have the capability to perform classical stability analyses.

The EASY program, originally developed by Boeing under Air Force Contract, is a general purpose program for the linear and nonlinear analysis of system dynamics using classical techniques. It has been used to model a variety of systems including environmental control systems, aircraft flight controls and dynamics, space vehicle dynamics, electrical power generation, rapid transit vehicles and air cushion landing systems.

The objective of this development effort was to develop an ejection seat classical stability analysis capability by incorporating SAFEST simulation subroutines into the EASY standard component library. The resultant computer program described in this User Manual/document is EASY And SAFEST Integration for the Evaluation of Stability and Trajectory (EASIEST).

Although EASY was originally developed under contract to the Air Force, additional Boeing funded research and development effort was undertaken to improve the program and increase its capability. The resultant improved version, EASY5, formed the basis for development of EASIEST. Because these added capabilities were developed using Boeing funds, they remain proprietary to the Boeing Company. Therefore, the program documentation/user manual is contained in two volumes. Volume I is a "stand-alone" user manual describing the EASIEST program characteristics and complete information on the use of the program and how to apply it to ejection seat dynamics and control analysis. It contains listings of the procedure files, models, analysis, standard components, and subroutines. Volume II is Boeing proprietary and contains only the source code listings of EASY5.

FOREWORD

This report describes research work performed by the Boeing Military Airplane Company, Seattle, Washington, for the Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratory, Wright-Patterson Air Force Base, Ohio, under Contract No. F33615-79-C-3407, Project 2402, "Vehicle Equipment Technology," work unit 24020328, "Ejection Seat Stability and Control Analytical Computer Program." Project engineer for the contract was Lanny A. Jines, AFWAL/FIER. This research work is part of an effort to develop an escape system computer simulation for performance analysis of ejection seat dynamics during escape. This report is in two volumes and combines the technical report and user manual. Volume I is the EASIEST "stand alone" user manual. Volume II contains the Boeing proprietary EASY5 source code. Volume II shall not be disclosed outside of Government agencies for a three-year period following completion of this contract and may be extended for an additional three-year period or successive three-year periods, by agreement between The Boeing Company and the Government.

The work reported herein was performed during the period of May 1979 to September 1980.

Roger F. Yurczyk served as the program manager. The technical work was performed by Christopher L. West and Brian R. Ummel, with consultation from John D. Burroughs.

TABLE OF CONTENTS

SECTION PAGE

I. INTRODUCTION 1

II. MODEL GENERATION 5
1. NAMING CONVENTION 6
a. Standard Component Naming Conventions 6
b. State, Variable, Parameter, and Table Naming Conventions 6
2. MODEL DESCRIPTION 9
a. Phrases and Delimiters 9
b. Command Phrases 12
3. MODEL SCHEMATIC 29
a. Standard Schematic Form 29
b. Input Quantity Labeling 29
c. Component Connection Paths 30
d. Additional Pages 30
e. Guidelines for Schematic Layout 30
4. WARNING MESSAGES 32

III. DYNAMIC ANALYSIS OF CONTINUOUS OR DISCRETE SYSTEMS 38
1. MODEL INPUT DATA 39
a. Scaler Data 39
b. Tabular Data 39
c. Matrix Data 43
2. INITIAL CONDITION, ERROR, AND INTEGRATION CONTROLS 44
3. INITIAL CONDITION COMMANDS 46
4. SIMULATION COMMANDS 47
5. PLOT DESIGNATION COMMANDS 52
6. STEADY STATE COMMANDS 55
7. LINEAR ANALYSIS COMMANDS 56
8. STABILITY MARGIN COMMANDS 58
9. FREQUENCY RESPONSE COMMANDS 59

TABLE OF CONTENTS (Continued)

SECTION		PAGE
	10. ROOT LOCUS COMMANDS	60
	11. EIGENVALUE SENSITIVITY COMMANDS	62
	12. FUNCTION SCAN COMMANDS	63
	13. OPTIMAL CONTROLLER DESIGN COMMANDS	64
	14. WARNING MESSAGES	69
	15. RENAMING MODEL INPUTS AND OUTPUTS	77
	16. COMPUTING TYPE ZERO TRANSFER FUNCTIONS WITH EASY	78
IV.	EASIEST STANDARD COMPONENTS AND SUBROUTINES	80
	1. STANDARD COMPONENTS	80
	2. SUBROUTINES	94
	3. MODELING WITH THE EASIEST COMPONENTS	94
	4. SEQUENCING AN EASIEST MODEL	99
V.	PROCEDURES FOR INSTALLING THE EASY5 PROGRAM AND SUBMITTING AN EASIEST RUN	101
	1. INSTALLING THE EASY5 PROGRAM	101
	2. SUBMITTING AN EASIEST RUN	103
VI.	EJECTION SEAT ANALYSIS EXAMPLE	105
VII.	CREATING AND MODIFYING STANDARD COMPONENTS	145
	1. MODIFYING THE FORTRAN SUBROUTINE OF AN EXISTING STANDARD COMPONENT	147
	2. MODIFYING THE RANDOM ACCESS FILE EZSTDBF	148
	3. CREATING A NEW EASIEST STANDARD COMPONENT	150
	4. LIBRARY EZSTLIB SIZE REDUCTION	151

TABLE OF CONTENTS (Continued)

SECTION	PAGE
VIII. DESCRIPTION AND GUIDE TO USE OF NUMERICAL INTEGRATION	152
1. CHANGES IN INTEGRATORS	152
2. GENERAL SELECTION GUIDELINES	153
3. ACCURACY AND ERROR CONTROL	156
4. STABILITY	159
IX. DISCRETE SYSTEM ANALYSIS TECHNIQUES	166
1. INTRODUCTION	166
2. SYSTEM EQUATIONS	166
a. Continuous System Stability Matrix	168
b. Discrete System Transition Matrix	169
c. Continuous System Transition Matrix	169
d. Calculation of Continuous System Transition Matrix	171
3. COMBINED SYSTEM TRANSITION MATRIX	172
a. Single Sample Rate	172
b. Integer Multiple Sampling Rate	175
c. Noninteger Multiple Sampling Rates	179
REFERENCES	182
APPENDIX A. MODEL GENERATION COMMANDS	183
APPENDIX B. ANALYSIS COMMANDS	185
APPENDIX C. CHECKLISTS	189
APPENDIX D. EASIEST INPUT/OUTPUT LISTS AND ASSOCIATED FIGURES	194
APPENDIX E. PROGRAM AEROMED	268
APPENDIX F. EASIEST PROCEDURE FILE	272
APPENDIX G. EASIEST STANDARD COMPONENTS	279
APPENDIX H. EASIEST SUBROUTINES	438

TABLE OF CONTENTS (Continued)

SECTION		PAGE
APPENDIX I.	FILOAD INPUT DATA	485
APPENDIX J.	EASIEST F-4E MANEUVERING COEFFICIENTS	499
APPENDIX K.	EASY5 INPUT/OUTPUT LISTS	510
APPENDIX L.	EASY PROGRAM ANALYSIS DESCRIPTION	597
APPENDIX M.	OPTIMAL CONTROLLER DESIGN WITH THE EASY PROGRAM	625
APPENDIX N.	EASIEST EXAMPLE	657

LIST OF ILLUSTRATIONS

FIGURE		PAGE
1	Character Assignment in Input/Output or Table Name	7
2	Model Generation Program Schematic	11
3	Typical Component Input/Output Configurations	15
4	Component Connection Paths	31
5	Model Generation Program Input File	106
6	EASY Schematic	108
7	Analysis Program Input File	112
8	Steady State Output	117
9	Simulation Output	124
10	Example of Printer Plots	141
11	Stability Regions for Runge-Kutta Methods; Orders 1-5	154
12	Stability Regions for Adams-Moulton Formulas	162
13	Stability Regions for Predictor - Corrector Pairs	163
14	Stability Regions for STIFF GEAR Formulas of Orders 1-3	164
15	Stability Regions for STIFF GEAR Formulas of Orders 4-6	164
16	Example of Continuous, Delay, and Sample States	167
17	Single Sampling Rate Example	173
18	Pictorial Representation of Single Sampling Rate Transition Matrices	173
19	Multisampling Rate Example	176
20	Pictorial Representation of Multisampling Rate Transition Matrices-Integer Multiple Rates	177
21	Pictorial Representation of Multisampling Rate Transition Matrices-Noninteger Multiple Rates.	180

LIST OF ILLUSTRATIONS (CONT'D)

FIGURE		PAGE
22	Standard Component "AP" Input/Output Overview	208
23	Standard Component "AS" Input/Output Overview	212
24	Standard Component "CT" Input/Output Overview	224
25	Standard Component "DR" Input/Output Overview	227
26	Standard Component "PC" Input/Output Overview	245
27	Standard Component "RL" Input/Output Overview	250
28	Standard Component "SP" Input/Output Overview	260
29	Standard Component "SP" Gimbal Spring and Vernier Rocket	261
30	Standard Component "SR" Input/Output Overview	264

LIST OF TABLES

TABLE		PAGE
1	System Model Description	10
2	EASY5 Program Language Delimiters	12
3	Integration Method Selection	48
4	Print Control Values	49

LIST OF ABBREVIATIONS

AFFDL	Air Force Flight Dynamics Laboratory
ASD	Aeronautical Systems Division
DART	Directional Automatic Realignment of Trajectory
EASIEST	EASY And SAFEST Integration for the Evaluation of Stability and Trajectory
EASY	Environmental control Analysis SYstem
SAFEST	Simulation and Analysis of In-Flight Escape System Techniques

SUMMARY

High performance combat aircraft have extended the maneuvering/operating range into regimes that exceed the capabilities of current ejection seat systems. One of the problems encountered involves the unstable rotational characteristics of the typical ejection seat, resulting in a decreased probability of survival due to the reorientation of the ejecting crewmember into an attitude less tolerant to acceleration. Furthermore, an unstable ejection seat may neither clear the airframe, nor provide adequate ground clearance. The capability to simulate the trajectory of an escape system, and to determine its stability characteristics using classical stability and control methods, is required to enhance the development of both active and passive stability augmentation systems.

The Simulation and Analysis of In-Flight Escape System Techniques (SAFEST) computer program, developed by the AFFDL for the analysis of occupied ejection seat stability characteristics, is a six-degree-of-freedom simulation of an ejection system. SAFEST uses a fourth order Runge-Kutta integrator with a fixed time step to calculate the trajectories for the seat/man, man alone, airplane, drag parachute, and the recovery parachute. However, SAFEST does not have the capability to perform classical stability analyses.

The EASY program, originally developed by Boeing under Air Force Contract, is a general purpose program for the linear and nonlinear analysis of system dynamics using classical techniques. It has been used to model a variety of systems, including environmental control systems, aircraft flight controls and dynamics, space vehicle dynamics, electrical power generation, rapid transit vehicles and air cushion landing systems.

The objective of this development effort was to develop an ejection seat classical stability analysis capability by incorporating SAFEST simulation subroutines into the EASY standard component library. The resultant computer program described in this user manual/document is the EASY And SAFEST Integration for the Evaluation of Stability and Trajectory (EASIEST).

Although EASY was originally developed under contract to the Air Force, additional Boeing funded research and development effort was undertaken to improve the program and increase its capability. The resultant improved version, EASY5, formed the basis for development of EASIEST. Because these added capabilities were developed using Boeing funds, they remain proprietary to The Boeing Company. Therefore, the program documentation/user manual is contained in two volumes. Volume I is a "stand alone" user manual describing the EASIEST program and complete information on the use of the program and how to apply it to ejection seat dynamics and control analysis. Volume II is Boeing proprietary and contains only the EASY5 source code.

SECTION I
INTRODUCTION

The objective of the research work described in this document was to develop a stability analysis capability for ejection seat performance. This was accomplished by modifying ejection seat simulation subroutines from an Air Force Flight Dynamics Laboratory (AFFDL) computer program, Simulation and Analysis of In-Flight Escape System Techniques (SAFEST), into a component library compatible with the EASY computer program. The resultant computer program described in this document has been termed the EASY and SAFEST Integration for the Evaluation of Stability and Trajectory (EASIEST).

Technology improvements in advanced combat aircraft have expanded the operational maneuvering envelope beyond the capability of current ejection seats. The aerodynamic instability of ejection seats during entrance into the air stream has led to tumbling, spinning, parachute shroud fouling, and a variety of system failures. The resultant loads may exceed the tolerance limits of the human body. Experience from combat aircraft involving fatalities and severe injuries points to the need for the development of stable ejection seats whose performance is designed to be within human tolerance limits.

The AFFDL has an active technology program to enhance the stability of an ejection seat. One aspect of the current technology has been the development of SAFEST, an escape system computer program for performance analysis of ejection seat dynamics. However, an ejection seat stability study utilizing the SAFEST program demands numerous simulation runs. The results obtained then require followup analytical data reduction to identify the system stability characteristics.

The EASY program was originally developed under an Air Force contract to provide methods for modeling and analyzing aircraft environmental control systems. In 1976, a second Air Force contract extended the application of the program to include aircraft flight dynamics. Since October 1976, a

Boeing-funded research and development effort has been undertaken to modify the program for use on a wide variety of control system analyses. Additional effort during the last half of 1977 and 1978 resulted in the development of the EASY5 program. The program now includes component models for many types of vehicles and control components, matrix and vector notation at all program levels, capability to model and analyze continuous and discrete systems, larger modeling capacity, and the ability to store time history data on magnetic tape, to name a few.

EASY5, with its additional capability, was used as the basis for the development of EASIEST. Because the advanced features of EASY5 were developed by Boeing-funded research, they remain proprietary to the Boeing Company. Therefore, the program has been documented in two separate volumes. Volume I is a complete "stand alone" user manual. Volume II is Boeing proprietary and contains only the listings of the EASY5 source code.

In the context of this document, EASY refers to the basic dynamics analysis program (Model Generation Program and Analysis Program) as developed under Air Force Contract F33615-76-C-3100 and modified under contract F33615-76-C-3165. EASY5 refers to the latest version of the EASY program which is Boeing proprietary. EASIEST refers to the standard components and algorithms developed specifically for ejection seat system analysis.

The EASY5 program is a user oriented computer program designed to provide a simplified way to describe and analyze linear and nonlinear dynamic systems. This simplified system description is then used for a wide variety of system analyses including conventional linear analysis and nonlinear simulation. The EASY5 computer program consists of a Model Generation Program and an Analysis Program. Both continuous and sampled data systems may be described and analyzed. The modeling of most of the systems is accomplished by describing the system in terms of standard components which are subroutines that model specific hardware items, like rate gyros, or perform certain functions such as wind gust generation. The models of these standard components have been constructed in a general fashion so that by proper choice of input parameters and tables, a wide range of

specific, required system components can be modeled by each standard component. If a portion of a particular system to be studied cannot be described by using one of the standard components, FORTRAN statements can be directly included in the model description to implement those portions of the system. Using a simplified description of the system model, the EASY5 Model Generation Program generates the required FORTRAN subroutines which accurately represent the model in program form. This computer generated model can then be analyzed by any of the nonlinear, linear, dynamic, or steady state evaluation techniques available in the EASY5 Analysis Program. The capabilities include the following:

- Algebraic sensitivity
- Eigenvalue and Eigenvalue sensitivity* determination
- Frequency response (Bode, Nyquist, and Nichols plots)
- Linear model generation
- Nonlinear simulation (time histories)
- Optimal control synthesis*
- Root locus*
- Stability margins*
- Stability matrix calculation
- Steady state analysis

*These analyses are not available for discrete systems.

Volume I of this document provides information on the use of the EASIEST program and how to apply it to ejection seat dynamics and control analysis. Section II of Volume I presents the details of how to use the Model Generation Program to construct a model. Section III presents the details of how to conduct a system analysis with the Analysis Program. It discusses how to input the model data, set initial conditions, designate plots and to select the different analysis options. Section IV describes the EASIEST components which are used to form the ejection seat dynamic models. Section V contains the procedure for program execution. Section VI presents an ejection seat analysis example. Section VII describes the procedure for the modification of a standard component. Section VIII contains a

discussion of the numerical integration options available. Section IX presents a discussion of the methods used for discrete system analysis.

Lists of Model Generation and Analysis Program commands for easy reference are available in Appendices A and B.

Appendix C presents a program checklist to help ensure that the program is being properly utilized.

Appendix D contains input and output tables for all the EASIEST standard components. Descriptive figures are also presented for the more complex standard components.

Appendix E contains the listing of program AEROMED, a postprocessor which calculates the aeromedical variables.

Appendix F contains a listing of the EASIEST procedure file.

Appendix G presents listings of the EASIEST standard components, and Appendix H contains associated subroutine listings.

Appendix I has the FILOAD input data. FILOAD is a program which creates a random access file from input data that defines the variable names on the calling sequence for each standard component.

Appendix J contains the EASIEST F-4E maneuvering coefficients for the airplane component.

Appendix K contains input and output tables for the EASY5 standard components developed under previous contracts.

Appendices L and M present descriptions of analysis calculations and optimal controller design, reproduced from Sections 4.4 and 4.5 of reference 1.

Appendix N presents a supplementary ejection seat analysis example.

SECTION II
MODEL GENERATION

The EASY5 Model Generation Program uses a block diagram type of approach for constructing the different system models. It is based upon the assumption that the system analyst will construct a detailed schematic block diagram of the system to be analyzed. This detailed schematic will then be changed to a form containing standard components FORTRAN. The parts of a system which cannot be modeled using these standard components are included by appropriate FORTRAN statements in the system description.

All interconnections between the different standard components and the aforementioned FORTRAN statements are accomplished by the Model Generation Program. The analyst draws the block diagram by specifying the location of each standard component or FORTRAN block in the schematic diagram and all of the components that provide inputs to that component. The Model Generation Program then generates name labels and the proper interconnections between the specified components. This is accomplished by matching the input quantities required by each component to the output quantities of the components specified as providing inputs.

After processing the complete system model description, the Model Generation Program generates the schematic diagram of the model showing all of the interconnections between the components in a mannner similar to the analyst's original diagram. It shows the quantities such as forces, moments, velocities, etc., that are used to form each interconnection. This schematic is produced on the lineprinter and provides a rapid graphic check on the program's interpretation of the model description.

In addition, the program produces a complete list of the input data that will be required by each component to complete the model description. The scalar and vector parameters and tabular data required for the analysis are included in this list. The program assumes that any quantity not supplied by another component will be supplied as a fixed parameter by the analyst.

Thus, requests for nonparameter items in the input data list reveal any connections that have been omitted from the system model description.

1. NAMING CONVENTION

Every variable or state must have a unique name. FORTRAN limits these to seven characters. For standard components, the name is associated with the standard component name.

a. Standard Component Naming Conventions

All standard components are given names consisting of two characters, the first of which is alphabetical. Thus we have LA for lag, CT for catapult, SL for sled, etc. A specific component in a model is distinguished from other components of the same type by adding one or two more characters to the standard component name. These characters are usually numeric but can also be alphabetical or blanks. For example, a model using ten of the same type may have these components designated as:

LA 1, LA 2, LA 3, ...LA10

If matrix component notation is used, a single component may be defined as:

LA 1, N=10

This results in a single component LA 1 with a 10 vector assigned to those inputs and outputs with variable array length capability.

b. State, Variable, Parameter, and Table Naming Conventions

A consistent approach has been taken to the naming of inputs and outputs for standard components. This convention is denoted by Figure 1. As described in the figure, the standard component name is shown as the fourth and fifth character of the total element name. The last two characters are used to distinguish between several of the same component. The first three characters are used to designate the inputs and outputs of the components. The specific names of the input and output quantities for the

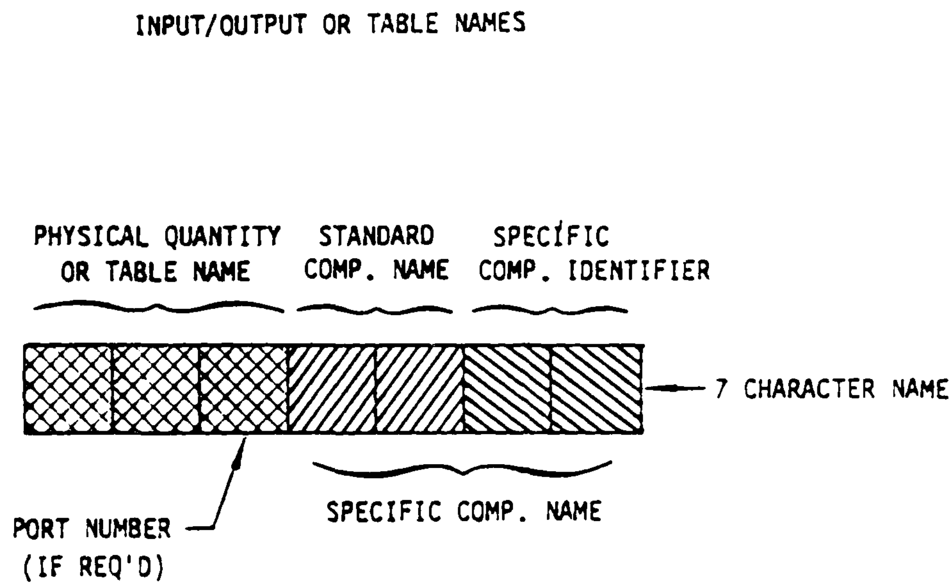


Figure 1. Character Assignment in Input/Output or Table Name

EASIEST components are listed on Appendix D. If a variable is a vector, subscripts must be added to the name when referring to a particular element in the array. An example of this would be S2 LA09 (2).

All of the input, output, and tabular quantities required by each component in a system model must have unique FORTRAN names. For standard components, these quantities are given names consisting of up to three characters that describe the physical quantity they represent. Since a single component may have several inputs or outputs of the same physical type, the program adds a "port" number as the second or third character of the physical quantity name to prevent such a duplication.

The physical quantities that are outputs of a standard component are specifically identified by adding the four character name of that component to the three character name of the physical quantity. In this way, unique seven character FORTRAN names are generated for all output quantities of the system model components. As an example, the output for standard component LA23 would be 52 LA23.

Input quantities to a component that are generated by another component carry the names of the component that generates them. Any inputs that are not satisfied by other model components are assumed to be parameters and are assigned the name of the component where they are an input.

If a component requires tabular data as an input, unique table names are generated just as scalar input quantity names by adding the component name to the table name. As an example, the input table for standard component SR11 would be TRFSR11.

All parameter, variable, and state quantities are set as real quantities even if their name starts with the FORTRAN integer letters I, J, K, L, M, N. Names added to the model via the ADD commands can consist of any valid FORTRAN name of up to seven characters. These names must not duplicate any name generated by the precompiler or other ADD statement.

2. MODEL DESCRIPTION

The Model Generation Program is a sophisticated precompiler which accepts model description instructions, and uses them to generate a FORTRAN model of the system. An EASY5 system model description contains numeric values, standard component names, and standard input and output quantity names. The instructions, referred to as "program commands," are made up of one or more functionally descriptive words.

The EASY5 commands may be best understood by using an example to describe a simple ejection seat model. The EASY5 system model description for it is given in Table 1.

As is seen in Table 1, the model description consists of a series of statements. Each statement specifies the location of each component in the schematic diagram and a list of all of the components that provide inputs to that component. The purpose of the location of the component in the schematic is to allow the Model Generation Program to use the line printer to draw a schematic of the model, such as shown in Figure 2. On the line printer drawn schematic, the input quantities to each component are shown. This can then be used to check functional flow for the diagram.

a. Phrases and Delimiters

The system model description is interpreted by the Model Generation Program from the command phrases following the program commands. The phrases must be separated by any one of the delimiter symbols shown in Table 2.

Comments can be inserted in the model description or analysis data by placing a "*" in column 1. These data cards will be ignored by the Model Generation or analysis programs.

TABLE 1 SYSTEM MODEL DESCRIPTION

```
                                   EASY5 MODEL GENERATION PROGRAM          VERSION 2.1.2

        ************************************************************************************************

        INPUT COMMANDS

COMMAND CARD ---> MODEL DESCRIPTION=MODEL CONTAINING AG,SL,RL,CT,SE,RS,AND CE COMPONENTS
COMMAND CARD ---> LOCATION=029    AG
COMMENT CARD ---> *
COMMAND CARD ---> LOCATION=027    FORT
COMMAND CARD --->                            ADD VARIABLES=CTFLAG
COMMAND CARD ---> FORTRAN STATEMENTS
COMMAND CARD --->       IF(TIME.LT.0.5) CTFLAG = 0
COMMAND CARD --->       IF(TIME.GE.0.5) CTFLAG = 1.0
COMMENT CARD ---> *
COMMAND CARD ---> LOCATION=022    SL
COMMAND CARD ---> LOCATION=025    CT      INPUTS=SL,SE(1=1),FORT(CTFLAG=SW)
COMMAND CARD ---> LOCATION=052    RL      INPUTS=SL,SE(1=1)
COMMAND CARD ---> LOCATION=057    RSCS    INPUTS=SE(SHP=XPB,UST=UPB,EST=EPB,WST=WPB)
COMMAND CARD ---> LOCATION=055    SE      INPUTS=RSCS(FPB=F2,2,TPB=T2,2)
COMMAND CARD ---> LOCATION=059    CE      INPUTS=RSCS
COMMENT CARD ---> *
COMMAND CARD ---> END OF MODEL
COMMAND CARD ---> PRINT
```

```
                   MODEL CONTAINING AG,SL,RL,CT,SE,RS,AND CE COMPONENTS                                   PAGE  0

1           2           3           4           5           6           7           8           9           10


11          12          13          14          15          16          17          18          19          20
                                WAPSL  =WAP
                                EAPSL  =EAP
       **********               UAPSL  =UAP**********              **********              **********
       *        *               XAPSL  =XAP*        *              *        *              *        *
       *  SL    *                          *  CT    *              *  FORT  *              *  AG    *
21     *   22   *=========================>*   25   *<=============*   27   *       28     *   29   *       30
       *        *                          *        *              *        *              *        *
       **********                          ********** CTFLAG =SW   **********              **********
           |                                   A
           |                        SRPSE =SRP |
           |                        USTSE =UST |
           |                        ESTSE =EST |
31         |2           33          34         |5           36          37          38          39          40
           |                        WSTSE =WST |
           |                                   |
           |                                   |
           |                                   |
           |                                   |
           |  WAPSL =WAP                       |
41         |2 EAPSL =EAP3           44         |5       WSTSE  =WPB     47          48          49          50
           |  UAPSL =UAP                       |  T11CT =T11
           |  XAPSL =XAP                       |  F11CT =F11
           V                    TM RL  =TM     |        ESTSE  =EPB             TRMRSCS=TRM
       **********               T21RL  =T21**********   USTSE  =UPB**********   TABRSCS=TAB**********
       *        *               F21RL  =F21*        *   SRPSE  =XPB*        *   FABRSCS=FAB*        *
       *  RL    *                          *  SE    *              *  RSCS  *              *  CE    *
51     *   52   *<=========================*   55   *<============>*   57   *=============>*   59   *       60
       *        *                          *        *              *        *              *        *
       **********SRPSE  =SRP               **********FPBRSCS=F22   **********              **********
                 USTSE  =UST                         TPBRSCS=T22
                 ESTSE  =EST
                 WSTSE  =WST
61          62          63          64          65          66          67          68          69          70


71          72          73          74          75          76          77          78          79          80
```

Figure 2. Model Generation Program Schematic

TABLE 2

EASY5 Command Phrase Delimiters

= equal sign

, comma

(left parenthesis

) right parenthesis

three or more blanks

b. Command Phrases

The EASY5 command phrases are described in this section. They are presented in a sequence similar to that in which they would be used in system model descriptions. For easy reference, they are listed at the end of this section in alphabetical order and in Appendix A.

MODEL DESCRIPTION

The MODEL DESCRIPTION program command is used to indicate the start of a new system model. This command may be followed, on the same line, by a title of up to 60 characters. This title will be used throughout the printout to identify various program output schematics and program listings. In the example shown in Table 1, the title is "MODEL CONTAINING AG, SL, RL, CT, SE, RS, AND CE COMPONENTS".

LOCATION

The LOCATION program command indicates the start of a new component in the system model. This command must be followed by a numeric value phrase that specifies the location of the component on the model schematic. Thus, in the example of Table 1, the location number of the component AG is 029 and component SE is 055, etc. To be a valid component location, the last two digits of this number must be a number between 1 and 80. The unit column of this number refers to a column on the schematic, while the tens column refers to a row. The hundreds column is used to specify additional pages,

if needed, for the schematic. Thus the numbers which would be valid location numbers for components on the first page, PAGE 0, of a system schematic are:

001, 013, 051, 080

These same locations on the second page of the schematic, PAGE 1, would be:

101, 113, 151, 180

The location number phrase is followed by the name of the component at that location. A LOCATION command must be given only once for each component. This means that once a LOCATION statement is started for a component, the complete description of that component must be given.

Certain components have variable length vectors associated with them. The number of elements in these vectors can be specified by providing a component dimension statement, N= or M=. Examples of this are:

```
LOCATION=002  LG 1   N=3       INPUTS=....

LOCATION=524  SM     N=12      INPUTS=....

LOCATION=913  IM     N=3,M=4   INPUTS=....
```

The N or M command must be the next command following the component name in the location statement. The phrase following the N or M command must be a number which specifies the dimensions of the arrays used by the component. The N or M commands can be applied to only those standard components which are designated to be capable of vector or matrix use as shown in the standard components lists contained in Appendix K. (None of the EASIEST components described in Appendix D require this command.)

Two characters immediately following the component name are used to designate multiple occurences of the same type of component within the model description. Thus the following are all valid component identifiers:

LG 1 LG15 LGIN LG2

This implies four occurences of the component LG.

Component arrays can also be identified in the same fashion.

LG1,N=3 LG15,N=4 LG2,N=5 LG,N=3

The above example identifies different distinctive lag filters with dimensions of 3, 4, 5, and 3 respectively. In each of the above examples, the Model Generation Program will use the blank space as a character in identifying the components. Thus LG 1 and LG1 are different components.

If a portion of a system cannot be conveniently modeled using standard components, a block of FORTRAN statements may be used. The location of the FORTRAN block in the system schematic diagram is specified by using the component name FORT. The use of this technique is described in the FORTRAN STATEMENTS section.

INPUTS

The INPUTS command indicates that the comma separated phrases following this command contain the names of the components that provide the necessary inputs to the component at that location.

In order to better understand the ways to connect component inputs and outputs, a description of these characteristics is needed. Figure 3 shows the three typical types of components and their connections. The first example in this figure shows an input/output configuration that has one input and one output, both designated by the letter S. Part 1 specifies the input, while part 2 the output. This type of component usually performs a mathematical operation. A second type of input/output configuration is also used for components that model specific physical items. For these components, the labels represent quantities that have a definitive meaning. Component TD in Figure 3 is an example of this. The input

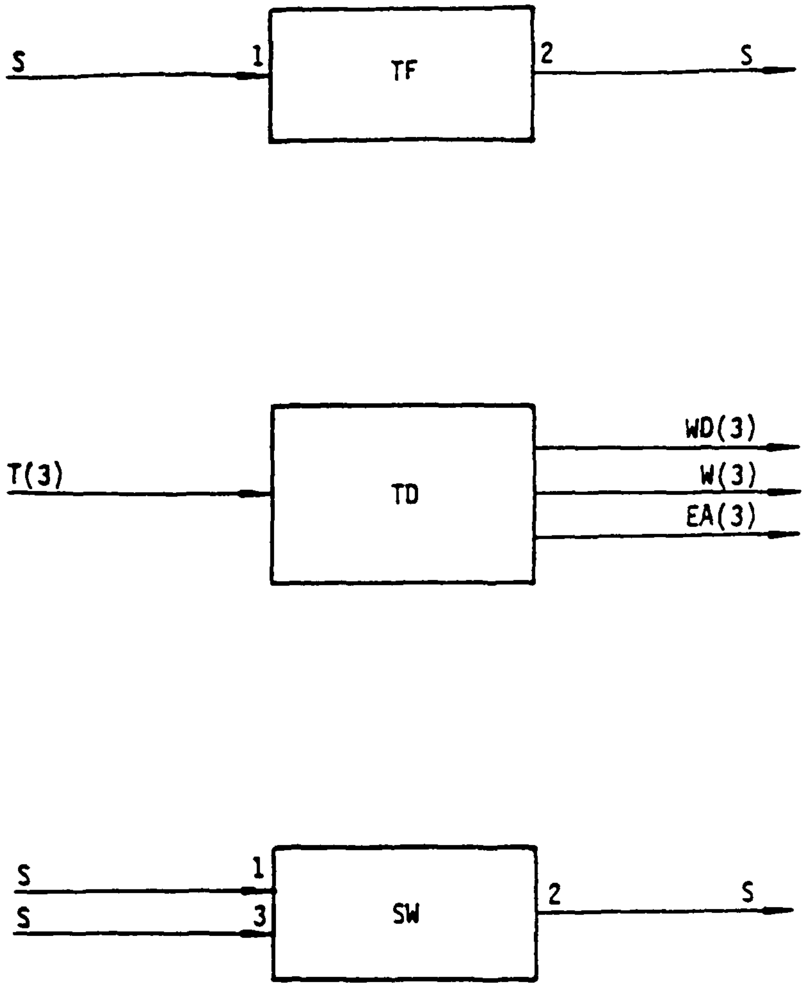


Figure 3. Typical Component Connections

quantity, T, is a vector which represents the torque applied to the vehicle. The output quantities are the vectors WD, angular acceleration, W, angular rate and EA, Euler angle. A third type of component has multiple inputs and/or outputs designated by S with a port associated with it. Component SW in Figure 3 is an example of this type. Extra care must be used defining the inputs and output connections to this device to assure proper signal hookup.

Between the components, three different levels of connection specification can be used in a model description:

1. Default (only component names are specified)
   Connections are made between all unconnected inputs and outputs to the first ports where a match of physical quantity names occurs. (Non-port inputs and outputs are also connected if a name match occurs.) For example:

   ```
   LOCATION = 045        SE        INPUTS = RL
   ```

2. Ports Specified
   Connections are made between matching physical quantities for all unconnected inputs and outputs of the specified ports. (Non-port inputs and outputs are also connected if a name match occurs.) For example:

   ```
   LOCATION = 045      SE      INPUTS = RL (1=1)
   ```

3. Physical Quantities Specified
   Connections are made between only those quantities specified. Previous connections cannot be over-ridden. For example:

   ```
   LOCATION = 045        RS        INPUTS = SE (SRP=XPB)
   ```

For many components, the input and output are single quantities and their connections can be made through the program default option without specifying the variable names. Thus, in the following example, component LG 1 at location 002 receives inputs from component MC 1:

```
LOCATION=002   LG 1   N=3   INPUTS=MC 1
```

In this example, the command phrase INPUTS is followed by a component name MC 1. The output name of MC and the input to the LG component have the same name, i.e., "S". Under this condition, no instruction other than specifying the input component is required.

For some components, there are multiple input and/or output "ports", which require the use of port numbers (S1, S2, S3, S4 etc.). The designation of these port numbers are defined in the standard components input/output lists. For multiple input ports, the port number must be specified as part of the INPUTS statement as shown in the following example:

```
LOCATION=110   MC 1    INPUTS=IT 1(S=S,1), TF 1(S=S,3)
```

It must be noted that <u>the output quantity comes first</u>, followed by the INPUT quantity name and port designation.

Port numbers refer to different physical connection points on a standard component. Once a connection is made between a port, such as port 2, of one component to another port, such as port 1, of a second component, inputs and outputs for ports other than 1 and 2 will not be connected even though they may have matching physical quantity names.

Some standard components can be used with variable dimensions. This feature allows the array length of a standard component with this capability to be specified. Thus, in the following example, the multiply and add component MC 1 and integrator components IT 1 and IT 2 are each defined to have three vectors as their inputs and outputs. The INPUTS function connects the three integrator outputs (IT 1) to the port 1 inputs and the integrator outputs (IT 2) to the port 3 inputs as shown in the following example:

```
LOCATION=052  MC 1   N=3   INPUTS=IT 1(S=S,1),IT 2(S=S,3)
```

```
LOCATION=032  IT 1   N=3   INPUTS=MC 1(S=S)

LOCATION=072  IT 2   N=3   INPUTS=MC 1(S=S)
```

If the input ports are not specified, the program default option will make the port selection in the order that they appear in the standard components list description. Thus, the following coding example would have accomplished the same objective.

```
LOCATION=052   MC 1   N=3   INPUTS=IT 1, IT 2

LOCATION=032   IT 1   N=3   INPUTS=MC 1

LOCATION=072   IT 2   N=3   INPUTS=MC 1
```

For certain components, such as control elements, the inputs to the component can be any physical quantity in the model. For these components, the input component names must be supplemented by the name of the particular output quantity that is to provide the input. As an example, consider a component that represents a linear first order lag transfer function. If the transfer function component's output, S, is to be the input torque, T, of the seat equations of motion, then the following statement would indicate to the program that, of the outputs of LG 1, S was to be used as the input, T, to the Seat Equations of Motion, SE 1:

```
LOCATION=005   SE 1   INPUTS=LG 1(S=T)
```

Input/output quantities may be either scalar, vector, or two dimensional arrays. Connections between array quantities are checked for compatible dimensions by the EASY5 Model Generation Program precompiler. An element of an output array can be used to drive a scalar input. Such a connection can be specified as:

```
LOCATION=043   LA   INPUTS=MM(A(2,3)=S)
```

Here A is a two dimensional array output by a component MM. Element 2, 3 of this array will drive input S of component LA. Numeric values following an output quantity array name are assumed to be element designations if enclosed in parenthesis. If any other delimiter is used, they are assumed to be port designations.

Inputs to standard components from FORTRAN blocks are provided by using the name FORT for the component name in the input expression, i.e.:

```
LOCATION=024    LA    INPUTS=FORT(COMP2(2)=S)
```

The FORTRAN component subscripted output quantity COMP2(2) will be connected to the input, S, of the standard component, LA. A discussion of using FORTRAN components is provided in the FORTRAN section. If a standard component is driven by both standard components <u>and</u> FORTRAN blocks, the standard component inputs must be specified <u>before</u> the FORT inputs.

Inputs to FORTRAN blocks may be either the outputs of standard components or the outputs of other FORTRAN blocks. Since the FORTRAN blocks do not have predefined input quantity names, the format used for specifying their inputs is different than that used for standard components. The complete name of the output quantities providing the inputs are required. The output names must contain enough information to uniquely define the source of the input. Thus, the complete output name of any standard component output must be given, i.e.:

```
LOCATION=63 FORT   INPUTS=S2 LA, PITCH, ROLL
```

Here the quantity S2 LA is the output of the standard component LA. PITCH and ROLL are the outputs of some other FORTRAN block. The above INPUTS statement refers to the output of the scalar LA component as S2 LA, not S,2 LA. The output quantity names must always be defined this way for use in FORTRAN component inputs since the EASY precompiler would interpret S, 2 LA as two separate input names.

FORTRAN STATEMENTS

The FORTRAN STATEMENTS program command allows the system analyst to supplement the standard EASY5 components with FORTRAN statements. Using this feature, the analyst can introduce his own program logic, DO loops, etc., as necessary to model any system not conveniently described with standard EASY5 components. Using this feature of the program, the analyst must perform many of the detailed connections and naming of variables that are normally accomplished by the EASY5 program. In return for these added tasks, the analyst gains a great deal of freedom and flexibility in forming details of his system model. To add a block of FORTRAN statements to the model, have it drawn on the schematic and included in implicit equation checking, the following convention must be used:

- A LOCATION statement with the component name FORT is placed before the FORTRAN STATEMENTS command. Input variables are specified by giving their names following the INPUTS command as described previously. These names may be either standard component output names or the outputs of other FORTRAN components, but must conform to the convention defined above.
- Outputs are specified by placing the ADD VARIABLES command <u>following</u> the INPUTS command. These quantities, either scalar or matrix or a combination, will be added to the model and assigned as outputs from the specific FORTRAN component. These output names may have up to seven characters.
- Parameter values, either scalar or matrix, are specified by the ADD PARAMETERS or ADD TABLES commands. These commands are added after the ADD VARIABLES command. These quantities will be added to the model and their values will be set in the Analysis Program. Parameter and table names may also have up to seven characters.

Thus the form for each FORTRAN component is:

```
LOCATION=063   FORT   INPUTS=S2 LA,ALPHA

ADD VARIABLES=BETA,GAMMA(3,3)

ADD PARAMETERS=COEFFS(3,2),GAIN

ADD TABLES=AEROTAB(250),3,AIRDATA(500),1

FORTRAN STATEMENTS
```

The lines before the FORTRAN STATEMENTS command (except ADD PARAMETERS and ADD TABLES) are required to specify the schematic location and the inputs and outputs to the block. If all of these are omitted, the FORTRAN statements will not appear in the schematic and will not be included in the implicit equation checking, which is described later under END OF MODEL. Only those quantities designated by ADD VARIABLES can be visibly connected to other standard components or FORTRAN blocks. The ADD commands are discussed next and details of the model schematic drawing appear in Section II.3. The ADD commands are used instead of dimension statements for the terms too be used in the FORTRAN statements. The FORTRAN statements can then include any FORTRAN IV required to describe the item being modeled. To simplify a number of these statements, a matrix arithmetic language has been developed which can be used within the FORTRAN statements to simplify the model description. A complete description of the matrix macro language is contained in Section IV.

<u>ADD VARIABLES</u>

<u>ADD PARAMETERS</u>

<u>ADD TABLES</u>

The ADD commands are used in conjunction with the FORTRAN STATEMENTS command to add variables, parameters, and tables that occur within the user supplied FORTRAN statements, to the EASY5 generated system model.

Quantities that are not specified by one of these commands cannot be accessed or manipulated by the EASY5 Analysis Program. See the examples in the FORTRAN section above for the proper order and use of the LOCATION, INPUTS, ADD, and FORTRAN STATEMENTS commands. Before discussing these commands, a few definitions of the terms are in order.

Variables: Variables are all dynamic time varying scalar or matrix quantities in the system model that are not states. In general, variables are related to states by fixed algebraic relationships.

Parameters: Parameters are constant scalar or matrix quantities in the system model. Parameters can be manipulated by the analyst to alter the system model. Default values are provided for certain parameters. The parameter values are set during the analysis option of the program.

Tables: Tables are constant nonscalar quantities in the system model. Tables are used to represent algebraic functional relationships with one, two or three independent variables. All table values are input as part of the analysis option of the program.

The format for the ADD commands is that the command is followed by one or more phrases that contain the names of the variables, parameters, or tables. These names must be unique. All parameter, and variable quanti ties are typed as Real quantities even if their name starts with the FORTRAN integer letters I, J, K, L, M, or N. Names added to the model via the ADD commands can consist of any valid FORTRAN name of up to seven characters. These names must not duplicate any name generated by the precompiler or another add statement. Variables or parameters may be scalar, vector, or two dimension arrays. The integrator components, IT or IN, should be used to define the state variables for the new component applications if additional states are required. The integrator components are straight forward in their use for adding new differential equations to be solved.

Matrix parameters are added to the model by placing dimension information, enclosed in parenthesis, after the parameter name, e.g.,

```
ADD PARAMETERS=ARRAY(3,6) COEF(6) . . .
```

Note: The (and) delimiters must be used to enclose dimension information. Dimensions must be between 1 and 99.

Matrix outputs are created by placing dimension information, enclosed in parenthesis, after the quantity names, e.g.,

```
ADD VARIABLES=VAR(3,2)
```

In addition to each table name, two numbers which specify the amount of storage to be allocated for the table and the number of independent variables must follow the table name. Thus to add three tables to a model, the instruction would be:

```
ADD TABLES=AEROTAB(120)2, TARGET(260)3, NOISE(500)1
```

This would add the two dimensional table AEROTAB with 120 words of storage; the three dimensional table TARGET with 260 words of storage; and the one dimensional table NOISE with 500 words of storage. The amount of storage is given by the formula:

where N= I + J + K + D

N= the total storage required by the table, in words.

I= the number of data points in the first independent variable table.

J= the number of data points in the second independent variable table. (J=0 if there is only one independent variable.)

K= the number of data points in the third independent variable table (K=0 if there are only one or two independent variables.)

D= the number of data points in the dependent variable table.
D=I if there is only one independent variable.
D=I*J if there are two independent variables.
D=I*J*K if there are three independent variables.

TABLE DIMENSIONS

The TABLE DIMENSIONS command can be used to specify Standard Component table dimensions. This is used when the default value for a Standard Component's table; as specified in the input/output lists, is too large or too small. This may be used as shown in the following example.

```
LOCATION=27   FV   INPUTS=LA1, LA2
TABLE DIMENSIONS=FTAFV=500
```

The TABLE DIMENSIONS command in this example would increase the data storage for table FTA of the component FV from the default value of 171 to 500 words.

O.C. INPUTS

O.C. OUTPUTS

The O.C. INPUTS, O.C. OUTPUTS, and other commands starting with the letters "O.C." are used to include an optimal controller in the system model. A complete description of the calculation methods and theoretical basis for the optimal controller are presented in Appendix N. An optimal controller is a general purpose control component which can have an arbitrary number of inputs and outputs. It is, therefore, necessary for the system analyst to specify the identity of each optimal controller input and output. This is done using the O.C. INPUTS and O.C. OUTPUTS commands rather than the INPUTS command that is used for the other components. Optimal controller inputs are output quantities, either variables or states, from components which are used to sense the response of the system being controlled. Optimal controller outputs are input quantities, either variables or

parameters, to components that serve as the actuators to the system being controlled.

O.C. CRITERIA

The O.C. CRITERIA command is used to specify those output quantities from the components that are to be used as the criteria for designing the optimal controller. These quantities are specified in the same format as O.C. INPUTS. If no O.C. CRITERIA are specified, the O.C. INPUTS are used as the design criteria. A complete discussion of the use of O.C. CRITERIA is given in Appendix M.

O.C. ORDER

The O.C. ORDER command can be used to specify the order of the optimal controller. If the optimal controller order is not specified, it will be taken as the order of the system model. This will result in a total system order, (optimal controller plus system model), that is twice the order of the system model. In most cases, such a high order optimal controller is unnecessarily complex and impractical. The O.C. ORDER is limited to values between zero and the system model order.

O.C. MODEL ORDER

The O.C. MODEL ORDER command can be used to specify that a model order lower than that of the given system model, be used for the optimal controller design. This command is used when optimal controllers are to be designed for high order systems. By using a lower order model, the computer memory requirements and computation time can be greatly reduced. A complete discussion of the use of reduced model orders is given in Section 4.4 of reference 1. This section is reproduced in Appendix N.

O.C. ANALYSIS

The O.C. ANALYSIS command is used to specify that computer memory requirements provided in the system need only be large enough for the analysis of

an optimal controller. The memory required to analyze a system with an optimal controller is considerably less than that required to do an optimal controller design. Thus, if the purpose of a run is to analyze the performance of an optimal controller which was designed on a previous run, the O.C. ANALYSIS command can be used to reduce computing costs and flow time.

END OF MODEL

The END OF MODEL command phrase indicates that model description has been completed and that the Model Generation Program should proceed with the generation of the model subroutines. As part of the subroutine generation, the model components are checked for implicit relationships. An implicit relationship occurs when a variable is used as an input to a component before it has been calculated. This can occur if a variable is used as an input to a component that preceeds the component that generates the variable. Implicit relations such as this can often be resolved by reordering the sequence of the components in the model. If such reordering occurs, a warning message is printed identifying the components affected. It is possible to create models in which the implicit relationships cannot be resolved by such a reordering. In this case, a warning message will be printed stating that analysis results will be invalid. The implicit relationship must then be resolved by changing this model. Changes such as placing an additional state in the implicit loop or solving the implicit relationship algebraically can be used.

PRINT

The PRINT command phrase causes the program to: (1) draw a schematic of the system model, as shown in Figure 2, (2) print a list of input requirements for the model; and (3) print a source listing of the FORTRAN subroutines that were generated for the model. The Model Generation Program then terminates.

LIST STANDARD COMPONENTS

The LIST STANDARD COMPONENTS command phrase causes the program to print a list of all standard components. For each standard component, lists of inputs, outputs, and tables for that component are provided. For each input, the physical quantity name and port number is given. For each output, the physical quantity name, port number, and the word STATE is given, if the quantity is a state. For each table, the table name, the number of independent variables and the default value for data storage is provided. This command is usually given as the first command of a model description and will result in a list of all standard component information as the first output from the Model Generation Program.

PRINT STATEMENTS

The simulation operation of the EASY5 Analysis Program has several print output options. Most of these, as described in Section III, consist of fixed formats such as: all states, all variables, or a user furnished list of variables. An additional option is to execute a set of user furnished print statements. *These print statements are specified as part of the* model description via the PRINT STATEMENTS command. The PRINT STATEMENTS command must be followed by valid FORTRAN statements. These statements will be executed only when the Analysis program PRINT CONTROL = 8 is specified along with the desired print output periods. In general, only FORTRAN PRINT, WRITE, and FORMAT statements would be included as PRINT STATEMENTS. However, other valid FORTRAN statements can be included if additional calculations or control logic is desired. Any state, rate, variable, or parameter in the model is available for use in the PRINT STATEMENTS. The PRINT STATEMENTS command can appear only once in a model, anywhere between the MODEL DESCRIPTION and END OF MODEL commands. An example of the PRINT STATEMENT command is given below:

```
PRINT STATEMENTS
WRITE (6,111) AMISS, XLOC, YLOC, TIME

111    FORMAT (MISS DISTANCE = *, G12.5, * AT XX =  *, G12.5,
       1 * AND Y = *, G12.5, 3X, * TIME = *, G12.5)
```

DEBUG

The DEBUG command may be used to place print statements between each Standard Component in the model. These print statements will be executed only when the PRINT command is given to the Analysis Program. The printout that occurs will be that specified by the PRINT CONTROL command. This command is very helpful in locating the cause of arithmetic errors in a model. This command should be placed before the END OF MODEL command. It should be removed from the model description once the model is free of arithmetic errors.

ALPHABETICAL LIST OF COMMANDS

ADD PARAMETERS
ADD TABLES
ADD VARIABLES
DEBUG

END OF MODEL
FORTRAN STATEMENTS
INPUTS

LIST STANDARD COMPONENTS
LOCATION
MODEL DESCRIPTION

O.C. ANALYSIS
O.C. CRITERIA
O.C. INPUTS
O.C. MODEL ORDER
O.C. ORDER
O.C. OUTPUTS

PRINT
PRINT STATEMENTS
TABLE DIMENSION

3. MODEL SCHEMATIC

The Model Generation Program produces a schematic diagram of the system being modeled. This schematic is generated on the line printer with the computer printout. Its purpose is to provide a means of rapidly locating errors in the model description.

In order to construct a schematic diagram in an efficient manner with a reasonable size program, it was necessary to establish some simple rules for symbol generation, component connection paths, and labeling. If these rules are kept in mind when laying out a schematic for the system, the EASY5 produced schematic will match that developed by the analyst. If the rules are violated by the analyst's schematic, the EASY5 schematic will still be correct but may contain some unusual component connection paths, and some labeling information may be overwritten.

a. Standard Schematic Form

The EASY5 schematic diagrams are produced on a standard 11" by 14" lineprinter page with 80 component locations per page. A standard form containing only the location numbers can be obtained by executing the EASY5 Model Generation Program with the single program command, PRINT. This form can then be reproduced and the copies used as forms for drawing system model schematics.

b. Input Quantity Labeling

The names of the physical quantities that are input to one component from another component are listed adjacent to the downstream component symbol. The physical quantity name, i.e., first three characters of the quantity being driven, is also given. These labels are placed near the connecting line that joins the two components. Since these names are composed of the physical quantity name and the name of the component that generates the information, the source of the input is evident from the name itself. Parameter and tabular inputs to a component are not shown on the schematic.

c. Component Connection Paths

In order to simplify the EASY5 schematic drawing subroutine, it was necessary to limit the types of connecting paths between components to a few basic routes. These paths are shown in Figure 4. Connections between components on the same horizontal or vertical line are straightforward. However, connections between components that do not share a horizontal or vertical line require at least a two segment path. These paths have been arbitrarily chosen to follow a clockwise route. It is, therefore, advisable that components that are on diagonal locations be placed in a clockwise sequence. If counterclockwise flow between components is necessary, it can be accommodated by placing the components on the same horizontal or vertical lines. The EASY5 schematic drawing subroutine does not go around components that are on a connection path. Such components are "run-over" by the connecting line.

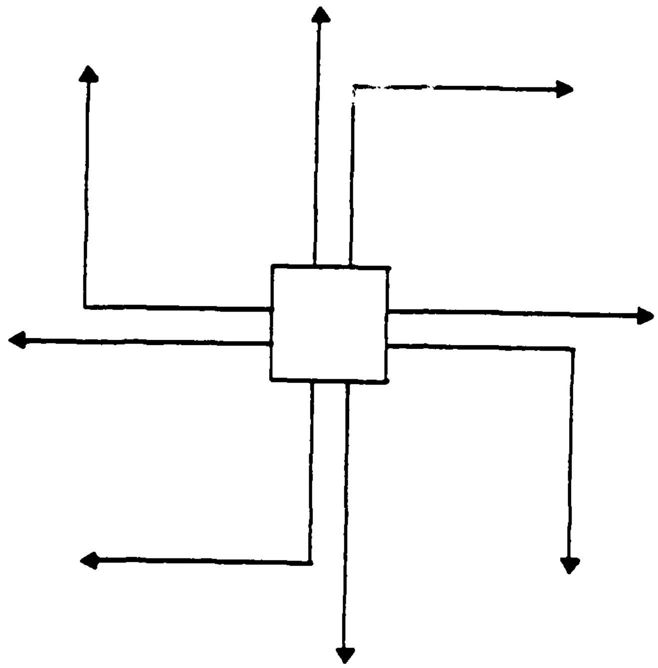
d. Additional Pages

The EASY5 schematic diagram may be broken down into as many pages as are necessary. No attempt is made to draw connecting paths between components located on different pages. It is, therefore, advisable to minimize the number of connecting paths between pages. This can usually be done by grouping components with many interconnections on the same page and placing page boundaries between such groups of components.

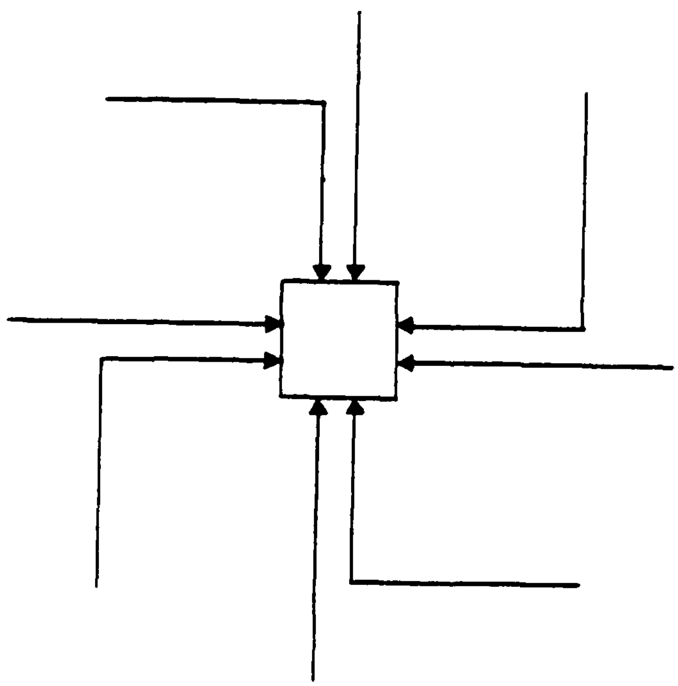
e. Guidelines For Schematic Layout

The following guidelines will help in creating schematic layouts that can be easily produced by the Model Generation Program.

- Try to place connected components on the same horizontal or vertical line.
- Avoid placing components on adjacent location points.
- Place diagonal components so that flow is clockwise.
- Group components to minimize flow paths between pages.



POSSIBLE OUTPUT PATHS



POSSIBLE INPUT PATHS

Figure 4. Component Connection Paths

4. WARNING MESSAGES

One or more of the following warning messages will occur if the program is unable to interpret a portion of the model description or encounters problems in assembling the system model. These messages will be preceded by: ***WARNING*** or ***NOTICE***. The symbols xxx and zzz are used to indicate phrases from the model description that are included as part of the warning message. The following messages are listed in alphabetical order:

1. ADD COMMAND MUST FOLLOW A "LOCATION=N FORT" COMMAND

The ADD VARIABLES command must follow a FORTRAN component location command.

2. CAN'T IDENTIFY SOURCE OF xxx INPUT TO LOCATION U

Cannot locate the source of xxx which is an input to component at location U.

3. CAN'T IDENTIFY xxx AS A STANDARD COMPONENT.

xxx will contain the first two characters of the phrase which cannot be identified as a command or standard component. This message will often follow other warning messages as the program makes successive attempts to interpret the given phrase.

4. CAN'T IDENTIFY xxx AS A VALID INPUT TO zzz

The input quantity xxx for component zzz cannot be identified.

5. CAN'T IDENTIFY xxx AS A VALID OUTPUT FROM zzz

The quantity xxx cannot be identified as an output from zzz.

6. CAN'T LOCATE FORTRAN COMPONENT xxx

Cannot locate FORTRAN component xxx statements.

7. CAN'T LOCATE O.C. INPUT, xxx, WILL RENAME AS: zzz

Check spelling of name xxx or that the quantity xxx has been renamed as a result of being driven by another component.

8. CAN'T LOCATE O.C. OUTPUT, xxx

Check spelling of name xxx.

9. COMPONENT xxx DEFINITION WASN'T COMPLETED BEFORE STARTING THE DEFINITION OF COMPONENT zzz

The command INPUTS was not given between the component names xxx and zzz. Check for proper spelling of INPUTS and a valid delimiter after the phrase xxx.

10. COMPONENT xxx HAS ALREADY BEEN DEFINED

11. CROSS PRODUCT IS ONLY DEFINED FOR 3 VECTORS

12. DIMENSIONS HAVE NOT BEEN GIVEN FOR xxx

Dimensions of input matrices must be defined before being used in a matrix expression.

13. DIMENSIONS OF xxx AND zzz ARE INCOMPATIBLE

Dimensions of input matrices in matrix expressions are incompatible.

14. DIMENSIONS OF xxx DO NOT MATCH THOSE OF zzz

Dimension mismatch occurred during interconnection of matrices.

15. LOCATION NO. xxx FOR COMPONENT zzz HAS LAST TWO DIGITS OUTSIDE THE ALLOWABLE RANGE 1 TO 80. NO SYMBOL WILL BE PLACED IN SCHEMATIC FOR THIS COMPONENT

This message will occur at the end of the model description for a component zzz which has an invalid location number. The system model may still be valid, but the schematic will not contain this component.

16. MATRIX xxx IS BEING DRIVEN BY A SCALAR QUANTITY zzz

This is likely to produce erroneous results.

17. MODES CANNOT BE SPECIFIED FOR COMPONENT xxx

The dimensions statements N=, M= can only be used on designated components.

18. NO OPTIMAL CONTROL INPUTS WERE SPECIFIED

Check that "O.C. INPUTS" command was used to specify optimal inputs.

19. NO OPTIMAL CONTROL OUTPUTS WERE SPECIFIED

Check that "O.C. OUTPUTS" command was used to specify optimal controller outputs.

20. NO xxx OUTPUTS MATCH UNSATISFIED zzz INPUTS

Check that it was intended to drive component zzz with component xxx or that the inputs to zzz have been previously satisfied by other component connections.

21. O.C. MODEL ORDER CANNOT BE SPECIFIED GREATER THAN MODEL ORDER

O.C. model order will be set to n.

22. O.C. ORDER CANNOT BE SPECIFIED GREATER THAN MODEL ORDER

O.C. order will be set to n.

23. ONLY 63 INPUTS + OUTPUTS ARE ALLOWED

Each component is limited to 63 inputs + outputs.

24. ONLY 100 VARIABLE DIMENSION COMPONENTS ARE ALLOWED

Only 100 variable dimension components are allowed in a given model.

25. SCALAR QUANTITY xxx IS BEING DRIVEN BY MATRIX zzz

The first element of matrix will be used to drive the scalar.

26. SYNTAX ERROR

Syntax error occurred in matrix expression.

27. TABLE NAME xxx MUST BE FOLLOWED BY A NUMERIC DIMENSION RATHER THAN zzz

When using the ADD TABLES command, it is necessary to provide the maximum amount of storage to be allocated for the table as well as the table name. This storage value must be a numeric quantity.

28. THE FOLLOWING COMPONENTS FORM AN IMPLICIT LOOP. MODEL RESULTS WILL BE INVALID. xxx, zzz, ...

Models must be explicit. Implicit loops can often be corrected by inserting a component with a state variable as its output, e.g., a simple linear lag, LA.

29. THE NUMBER OF O.C. INPUTS, OUTPUTS, OR CRITERIA VARIABLES MUST BE 63 OR LESS XXX WILL NOT BE LOADED

30. THE SEQUENCE OF THE FOLLOWING COMPONENTS HAS BEEN ALTERED TO FORM AN EXPLICIT MODEL. xxx, zzz, ...

The model component sequence as given contained an implicit relationship. By altering the component sequence, it was possible to form an explicit model.

31. xxx IS NOT A VALID DIMENSION

The phrase xxx should be numeric to be a dimension phrase.

32. xxx IS NOT A VALID INPUT QUANTITY OR PORT DESIGNATION FOR COMPONENT zzz

The phrase xxx cannot be located as one of the input quantities or input ports of the component zzz. No connections will occur. Check the list of standard components for the proper spelling or port designations for this component.

33. xxx IS NOT A VALID LOCATION NUMBER

The LOCATION command must be followed by a numeric location number.

34. xxx IS NOT A VALID PORT DESIGNATION FOR INPUT COMPONENT zzz. ERRONEOUS CONNECTIONS MAY OCCUR.

The phrase xxx cannot be located as a valid input port for the component zzz. Connections will be attempted using the upstream output port that was identified.

35. xxx IS NOT A VALID SUBSCRIPT

Subscripts must be numeric. The use of parenthesis as delimiter after array name implies a subscript is given.

36. xxx IS NOT A VALID SUBSCRIPT FOR FORTRAN OUTPUT zzz

The quantity xxx is not a valid subscript for FORTRAN output quantity zzz.

37. xxx IS NOT AVAILABLE AS INPUT

Cannot locate xxx as FORTRAN input to standard component.

38. xxx ISN'T NUMERIC O.C. ORDER MUST BE NUMERIC QUANTITY.

39. xxx MUST BE A SQUARE MATRIX

Simultaneous equation solution is valid only for square coefficient matrix.

SECTION III
DYNAMIC ANALYSIS OF CONTINUOUS OR DISCRETE SYSTEMS

The EASY5 Analysis Program allows several different dynamic, static, linear, or nonlinear analysis techniques to be used on the dynamic system model generated by the Model Generation Program. In addition to normal analysis techniques, optimal linear controllers based on Kalman optimal linear regulator and Kalman filter theory can be synthesized by the program. The performance of such optimal controllers when operating with the nonlinear system can be analyzed using any of the analysis techniques.

Both continuous systems, i.e., those described by ordinary nonlinear differential equations, and discrete systems, i.e., those described by differential and discrete difference equations, can be modeled and analyzed by the EASY5 program. The analysis techniques automatically switch to discrete methods* if one of the discrete components, DE, DF, DL, DT, DZ, or SH is included in the system model. All data input, output, and analysis commands are the same for both continuous and discrete systems. The only restriction for discrete systems is that the total number of sampling periods is restricted to 10.** This refers to the sampling period parameters, TAU, for each discrete component. The name of these parameters must always start with the letters TAU, and no other parameter may start with the letters TAU.

A description of the control of the program and of the analytical methods is given in Sections III.1 through III.16. An alphabetical listing of the analysis program commands is given in Appendix B of this document. Check lists for each analysis are given in Appendix C. For a description of continuous system techniques and numerical methods, see reference 1, Section 4. For discrete methods, see Section IX.

---

*The Root Locus, stability margin, eigenvalue sensitivity, and optimal controller design options are not available for discrete systems.

**Sample periods must be integer multiples of one another.

1. MODEL INPUT DATA

A dynamic system model requires that the values of numerous model parameters, tables and initial conditions, be provided to complete the model description. Sections III.1, III.2, and III.3 describe the methods used to specify parameter values, tables, and matrices.

a. Scaler Data

PARAMETER VALUES

This program command allows the numeric values of parameters to be loaded into the system model. The PARAMETER VALUES command is followed by one or more parameter names followed by a numeric value of ten characters or less. Each name and its value are separated by commas or another one of the standard delimiter symbols. This command is used to specify the values of all system model parameters at the beginning of an analysis. It may also be used at any point between analyses to modify the value of one or more model parameters. A default value of .99999 is provided for all parameters not specified.

PARAMETER VALUES = MASS = 10., AREA = 50, SW AG = 1,
CCGSE=.48,0,-.75, CW SE=210, STIPC=10.57,....

b. Tabular Data

TABLE

If tabular data is required by the system model, it should be loaded with the other parameter values before any of the analysis commands described in Sections III.4 to III.13 are issued. Tables may be modified between analyses by loading new values. The tables required by an EASY5 generated model are specified in the Model Generation Program Input Requirements List. These tables may have either one, two, or three independent variables. All data items are in a free field format with each item having

10 characters or less separated by commas or other standard delimiter. The data items required for each table are placed in the following format:

Line 1	TABLE	Table name	NX	NY	NZ
Line 2*	Z table values				
Line 3*	Y table values				
Line 4*	X table values				
Line 5*	D table values				

For this input, the following definitions apply:

Table Name	-	The seven character table name generated by the EASY Model Generation Program.
NX	-	The number of points in the first independent variable table.
NY**	-	The number of points in the second independent variable table.
NZ***	-	The number of points in the third independent variable table.
Z table***	-	Table of NZ third independent variable values.
Y table**	-	Table of NY second independent variable values.
X table	-	Table of NX first independent table values.
D table	-	Tables of dependent variable values.

---

*As many lines or cards as required may be used. Each table must start with a new line or card and NZ, NY, NX, and NX*NY*NZ points must be given per table.

**These items are omitted for tables with one independent variable.

***These items are omitted for tables with one or two independent variables.

A copy of all tabular input data is printed as it is interpreted from the data, unless the OMIT TABLE PRINTOUT command has been given. The following example shows the data for a one and a two independent variable table.

```
Line 1    TABLE, TAB-ONE,   10
Line 2    1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Line 3    11, 12, 13, 14, 15, 16, 17, 18, 19, 110
Line 4    TABLE, TAB-TWO, 5, 4
Line 5    10.3, 20.4, 30.5, 40.6
Line 6    1, 2, 3, 4, 5
Line 7    11, 12, 13, 14, 15
Line 8    21, 22, 23, 24, 25
Line 9    31, 32, 33, 34, 35
Line 10   41, 42, 43, 44, 45
```

The printout of these tables would be:

```
                              TABLE TAB-ONE

                    FIRST INDEPENDENT VARIABLE TABLE
1.000   2.000   3.000   4.000   5.000   6.000 7.000   8.000   9.000   10.00

                        DEPENDENT VARIABLE TABLE
11.00   12.00   13.00   14.00   15.00   16.00 17.00   18.00   19.00   110.00


                              TABLE TAB-TWO

                    SECOND INDEPENDENT VARIABLE TABLE
10.30        20.40          30.50          40.60

                    FIRST INDEPENDENT VARIABLE TABLE
1.000        2.000          3.000          4.000          5.000
```

DEPENDENT VARIABLE TABLE

11.00	12.00	13.00	14.00	15.00
21.00	22.00	23.00	24.00	25.00
31.00	32.00	33.00	34.00	35.00
41.00	42.00	43.00	44.00	45.00

THREE INDEPENDENT VARIABLE TABLE EXAMPLE

Line 1	TABLE=FTAFW	3	2	2
Line 2	1,2			
Line 3	3,4			
Line 4	5,6,7			
Line 5	111,112,113			
Line 6	121,122,123			
Line 7	211,212,213			
Line 8	221,222,223			

The printout of this table would be:

==== TABLE FTAFW ====

THIRD INDEPENDENT VARIABLE TABLE

1,000 2,000

SECOND INDEPENDENT VARIABLE TABLE

3,000 4,000

FIRST INDEPENDENT VARIABLE TABLE

5,000 6,000 7,000

DEPENDENT VARIABLE TABLE

THIRD INDEPENDENT VARIABLE = 1,000

111.0	112.0	113.0
121.0	122.0	123.0

THIRD INDEPENDENT VARIABLE = 2,000

211.0	212.0	213.0
221.0	222.0	223.0

OMIT TABLE PRINTOUT

The OMIT TABLE PRINTOUT command may be used to suppress the printback of table data. This command is often used on production runs or models with large amounts of constant tabular data. A second occurrence of this command causes table printback to be restored.

c. Matrix Data

Matrix Parameters can be one or two dimensional arrays. The matrix input format must contain the matrix name, the input method, and the appropriate matrix elements. If the input method is not specified, a default of input by columns is assumed. If the default mode is used, however, the user must be careful to:

- Input the exact number of elements defined by the dimensions in the Model Generation Program since the maximum dimensions are not checked by EASY5. With this method, the user must accept this responsibility.

- Not exceed ten characters per matrix element.

If the default option is not used, parameter arrays can be loaded by any of the following conventions after inserting the PARAMETER VALUES command:

COLUMN INPUT

ADATA, C (1, 1) 1, 2, 3, 4, 5	Starts at element 1, 1
ADATA, C (1, 2) 6, 7, 8, 9, 10	Starts at element 1, 2

ROW INPUT

BDATA, R (2, 3) 7, 8, 9, 10	Starts at element 2, 3
BDATA, R (1, 2) 3, 6, 9, 10	Starts at element 1, 2

DIAGONAL INPUT

COEF, D (2, 4) .3, .4, .5	Starts at element 2, 4

ZERO Array - then load by row

COEF, Z, R (2, 2) 1, 2, 3

Set array to infinite, "Infinite" = 10^{36}

COEF, I

Input by Column starting at element 1, 1 (default option)

VECTOR = 1, 2, 3, 4, 5

ELEMENT Input

ADATA (1, 2) = 12, (3, 4) = 16, (2, 3) = 21

Note: "(" must be used as delimiter immediately following array name.

2. INITIAL CONDITION, ERROR, AND INTEGRATION CONTROLS

<u>INITIAL CONDITIONS</u>
<u>ERROR CONTROLS</u>
<u>INT CONTROLS</u>

These program commands may be used to specify integrator initial condition values, error controls, or status, whether active (= 1) or frozen (= 0). The default values that are provided are 0.0 for initial conditions, 0.001 for error controls, and 1 for integration controls. These are furnished by the EASY5 Analysis Program. However, it is strongly recommended that values appropriate to the particular system model be furnished for the initial conditions and error controls.

Each of these commands is followed by phrases of the form of a state name followed by a numeric value. State quantities that are vectors or matrices may be input by the same conventions as for parameters. The following shows an example of how these commands are used:

```
INITIAL CONDITIONS = VELOCITY = 50.,  ANGLE = 2.,  U SD = 512,  362,0.
ERROR CONTROLS     = VELOCITY = .1,  ANGLE = .01,  U SD(3) = .0001
INT CONTROLS       = VELOCITY = 0,  ANGLE = 1,  STROKE = 1
```

ALL STATES
NO STATES

These program commands may be used to activate or freeze all system integrators. These commands are normally used together with the INT CONTROLS command to specify the desired integrator configuration.

INITIAL TIME = t

This program command allows the initial value of time to be specified. The default value of initial time is zero. The INITIAL TIME command is used with models that contain time dependent features where it may be desirable to have time at the beginning of a simulation run or during a steady state analysis be some value other than zero.

PRINT

This command, PRINT, causes the states to be set to the initial conditions, time to equal INITIAL TIME, and the model executed and printed output requested via the PRINT CONTROL command.

3. INITIAL CONDITION COMMANDS

<u>XIC-X</u>
<u>XIC-XIC1</u>
<u>XIC-XIC2</u>
<u>XIC-XIC3</u>
<u>XIC1-XIC</u>
<u>XIC2-XIC</u>
<u>XIC3-XIC</u>

These program commands are used to transfer data from the current state vector, X, to the initial condition vector, XIC, and between the XIC vector and three auxiliary initial condition vectors XIC1, XIC2, XIC3. The following shows how these commands would be used:

XIC1-XIC, XIC-X, XIC2=XIC

The three program commands shown above would take the current operating point (initial condition vector) and store it in vector XIC1; then transfer the current state, X, into XIC; and then store that value of XIC in XIC2.

<u>CALC XIC</u>

This command allows initial conditions to be calculated from manually input parameters or initial conditions. This command, CALC XIC, causes the state to be set to the values input manually for XIC; an integer flag in common block /CICCAL/ to be set to 1, and the model to be executed. Initial condition calculations can be placed in the model that will be executed only if the flag equals 1. Upon exiting from the model, the initial condition array XIC is set equal to the state array X and the print routine is called. The initial condition flag is reset to 0.

4. SIMULATION COMMANDS

SIMULATE

This program command initiates simulation operation. Before the simulate command is used, the following program values must be set:

TINC	=	time increment, seconds
TMAX	=	duration of the simulation run, seconds
INT MODE	=	integrator mode control
OUTRATE	=	output rate
PRATE	=	print rate
PRINT CONTROL	=	print control variable

These program commands specify the integration time increment, duration of simulation run, the integration type, the simulation output rate, the printing rate, and the quantity of printing, at each point in time. These quantities must be specified before the first use of the SIMULATE command.

For discrete systems, the time increment, TINC, should be an integer submultiple of the sample periods. Thus, if sample periods were .01 and .04, TINC should be selected such that: n*TINC=.01, where n is an integer. The EASY5 Analysis Program will check TINC and adjust it if necessary to satisfy this requirement. The output control OUTRATE will also be adjusted to maintain approximately the same data output rate.

The integration mode control, INT MODE, allows one of six different integration methods to be selected according to the description given in Table 3. The default value of INT MODE is 6. A description of tnese integration methods and a guide to their use is given in Section VIII.

TABLE 3

Integration Method Selection

INT MODE	Method
1	Variable Step, Variable Order Gear
2	Variable Step 4th Order Runga-Kutta
3	Fixed Step Huen Method, 2nd Order
4	Fixed Step Euler, 1st Order
5	Adams-Bashforth predictor/Adams-Moulton Corrector, Orders 2-12.
6	Stiff Gear

The time increment, TINC, provides the integrator time step size, in seconds, for the fixed step integrators. TINC also provides the report interval for which data will be available for printing or plotting. The default value for TINC is 0.1.

The duration of a simulation calculation is specified by the TMAX parameter in seconds. The default value of TMAX is 1.

The output rate parameter, OUTRATE, determines the rate at which simulation data is added to plots. Thus, if OUTRATE is set equal to 10, data will be plotted every 10th time increment, TINC. This feature is normally used only when a fixed step size integrator is specified. With such an integrator, the time increment is usually quite small, and excessive plotted output would be generated if it were not for this sampling feature provided by the OUTRATE parameter. The default value of OUTRATE is 1. OUTRATE should only be set to positive integer values.

The number of data samples plotted for a simulation analysis is given by:

$$\text{No. of Plotted Samples} = \frac{\text{TMAX}}{\text{TINC*OUTRATE}} + 1$$

For most simulation operations, the plot output is the primary data. The line printer output options provided by the PRINT CONTROL parameter allow a wide range in the amount of detailed information about the simulated system to be printed. The value of the PRINT CONTROL parameter sets the quality of data printed at each print report interval according to Table 4. Options 1 through 4 give "snap-shots" of all states, rates, variables, and parameters of the system model at a particular point in time. Option 5 provides tabular lists of up to 40 specified quantities. Options 6 and 7 are used with the steady state analysis options. Options 6 and 7 are used with the steady state analysis options. Option 8 uses the user provided print statements from the model description. The default value for PRINT CONTROL is 0.

TABLE 4

Print Control Values

PRINT CONTROL	Resultant Lineprinter Output
0	None
1	All states, rates, and time
2	All states, rates, variables, and time
3	All states, rates, variables, and parameters at time = 0
4	All states, rates, variables, and parameters
5	Time and the quantities spoecified via PRINT VARIABLES command
6	All states, rates, variables, and parameters at each STEADY STATE iteration
7	All states, rates, variables, parameters, and system Jacobian matrix at each STEADY STATE iteration
8	User furnished PRINT STATEMENTS (See Model Generation Section II.2.b)

The PRATE parameter determines the sampling rate at which the simulation data specified by the PRINT CONTROL parameter is presented on the lineprinter. Thus, if PRATE is set equal to five, data will be printed on the

line printer every fifth time it is added to the output plots. The default value of PRATE is 1. PRATE should only be set to positive integer values.

The number of data samples printed for a simulation analysis is thus given by:

$$\text{No. of Plotted Samples} = \frac{\text{TMAX}}{\text{TINC*OUTRATE* PRATE}} + 1$$

An example of the use of these commands is shown below:

PRINT CONTROL = 3, TINC = .01, TMAX = 10.,

INT MODE = 3, OUTRATE = 10, PRATE = 10, SIMULATE

In the example, the fixed step Huen integration method would be used with a step size of .01 second. The simulation would run for 10 seconds. Plotted output would occur every .1 seconds, (10*.01), and printed output would occur every 1.0 seconds (10*10*.01).

<u>TINC2</u>
<u>OUTRATE2</u>
<u>PRATE2</u>
<u>PRINT2</u>
<u>PRINT2 FROM</u>, t_1, <u>TO</u>, t_2

For some applications, a single set of output controls is not satisfactory. For example, it might be desirable to have a high sampling rate during an initial transient followed by a slower sampling rate, or to have a high sampling rate around a critical event. To satisfy this requirement, a second set of control values can be assigned to the program values TINC, OUTRATE, PRATE, and PRINT CONTROL. These are specified as:

TINC2, OUTRATE2, PRATE2, PRINT2

These values can be requested during a time interval via the command:

PRINT2 FROM, t_1, TO, t_2

Here t_1 is the time to start the second output option and t_2 is the time to revert to the original output option as given by: TINC, OUTRATE, PRATE, and PRINT CONTROL. An example of the analysis commands for this type of operation is:

```
PRINT CONTROL = 4,  TINC = .01,  TMAX = 10
OUTRATE       = 10,  PRATE = 10,  OUTRATE2 = 1,
PRINT2        = 8,  PRINT2 FROM,  8.,  TO,  9.,  SIMULATE
```

In the example, the simulation would run for 10 seconds with a step size of .01 seconds. The initial plotted output would be every 0.1, (10*.01), seconds and printed output would occur every 1., (10*10*.01) second. Between 8 and 9 seconds, the plotted and printed output rates would be increased to every .01, (1*.01), and 0.1, (10*1*.01) seconds and would consist of model furnished PRINT STATEMENTS (print option 8).

The second output options can also be activated by events occurring within the model. This can be done by setting a print flag variable, PFLAG, within the model EQMO subroutine to a non-zero value. As long as PFLAG has a non-zero value, the second output options will be in effect. When PFLAG is set to zero, the original output options are restored. PFLAG can be set by an IF test contained in a FORTRAN STATEMENT in the model. An example of this type operation is:

```
FORTRAN STATEMENT
     PFLAG = 0
     IF (RANGE .LT. 100.) PFLAG = 1
```

In this example, if the variable range becomes less than 100, the second print option will occur.

PRINT VARIABLES

This program command allows up to 40 variables to be specified for printing using option 5 of the PRINT CONTROL. This command is followed by from one to 40 state, rate, variable scalar, or subscripted names separated by delimiters. This command deletes all previously stored PRINT VARIABLES names. A column format will be used if the number of quantities being printed is less than or equal to 10. If more than 10 quantities are specified, the name and value of each scalar or subscripted vector quantity will be printed in a format similar to that of print options 1, 2, or 3. An example of this use is:

PRINT VARIABLES = S1 DE1, S1 DE2, W1 DE2, S2 LA(3)

5. PLOT DESIGNATION COMMANDS

DISPLAY1
DISPLAY2
DISPLAY3
DISPLAY4
DISPLAY5
DISPLAY6

These program commands are used to define the quantities to be displayed by off-line plots or written on external tapes for simulation or steady state calculations. These commands must be issued before the simulation or steady state analysis is requested. From one to five plots may be specified per display. Each plot is specified by stating the dependent variable and the independent variable separated by the letters VS. If desired, the dependent and independent axis scale ranges can also be specified. These scales will be used if the MANUAL SCALES commands are given. The independent scale range is specified by the word XRANGE followed by the minimum

and maximum values for this scale. The dependent scale similarly is spceified by the word YRANGE. If scale ranges are not specified, values will be used that span the given data. For more than one plot on a page, a common independent variable must be used.

The following example shows two ways to specify plots:

```
DISPLAY1
ANGLE, VS, TIME, YRANGE = -2,4
STROKE, VS, TIME, YRANGE = -.5,.5
P1 DE1, VS, TIME, YRANGE = 0,60
DISPLAY2
P1, CE, VS, TIME, YRANGE = -20,20
P1, DE2, VS, TIME, YRANGE = -15,15
PRESSURE, VS, TIME, YRANGE = -100,100
THECE, VS, TIME, YRANGE = -5,5
DISPLAY3
STROKE, VS, PRESSURE, YRANGE = -1.5, XRANGE = 300,500
```

<u>SI MANUAL SCALES</u>
<u>SS MANUAL SCALES</u>
<u>SI AUTO SCALES</u>
<u>SS AUTO SCALES</u>

The SI MANUAL SCALES and SS MANUAL SCALES commands allow the plotted output requested by the DISPLAY commands to be plotted on manual scales specified by the YRANGE and XRANGE commands. If manual scales are requested, manual scales <u>must be given</u> and will be used for <u>all</u> plots. The SI prefix is for simulation data and the SS is for steady state analysis. The SI AUTO SCALES and SS AUTO SCALES commands can be used to return plotting to the automatic scaling mode. Auto Scales are selected so that they span each plotted quantity. The auto scale option is the default used until manual scales are requested.

<u>PLOT ON</u>
<u>PLOT OFF</u>

These program commands allow the plotted output to be turned on or off. The default condition is PLOT OFF. It is, therefore, necessary to include the PLOT ON command before requesting any analysis from which plots are desired. The PLOT OFF and PLOT ON commands can be issued between analysis requests if it is desired to omit the plotting of certain analysis results.

OMIT PLOT POINTS

Boxes are normally drawn around each plotted data point. This command supresses these boxes. A second occurrence of this command restores the boxes around plotted data points.

CALCOMP
PRINTER PLOTS
SC4020
MTS PLOTS

Plots are routed to a particular physical device by specifying the above commands prior to the analysis which generates plotted data. Printer plots, MTS plots, and either CALCOMP or SC4020 plots may be generated in the same run.

PLOT
ID
TITLE

The PLOT ID program command allows an identification label to be placed as the first page of plotted output. Up to 48 characters may follow the delimiter that follows the PLOT ID command. This identification must be used to place mailing information on the plotted output.

The TITLE command allows a common title to be placed on all plotted output. Up to 74 characters may follow the delimiter that follows the TITLE command. The TITLE command may be changed before each analysis. Once

defined, the title remains in effect until a new title is entered. Examples of these commands are shown below:

```
PLOT ID = EX USER **M/S 70-16**
TITLE   = FLEX MODE CASE
```

6. STEADY STATE COMMANDS

<u>STEADY STATE</u>

This program initiates the calculation of the system steady state. Associated with this command are the program name and values:

1. SS PARAMETER = steady state parameter.
2. SS START = initial value of steady state parameter.
3. SS STOP = final value of steady state parameter.
4. SS POINTS = number of values the steady state parameter takes going from SS START to SS STOP.
5. SS ITERATIONS = maximum number of iterations allowed per steady state calculation.
6. PRINT CONTROL = print control variable.

SS PARAMETER specifies the parameter to scan from the value SS START to SS STOP in SS POINTS steps. SS ITERATIONS specifies an upper limit on the number of iterations to be used to calculate a steady state. The default value of SS ITERATIONS is 30. If the SS PARAMETER is blank, a single steady state calculation will occur. The steady state parameter can be any valid parameter name.

The PRINT CONTROL parameter provides all the print control functions described in Section III.4 for simulation operation plus two extra forms, 6 and 7, which may be used to track the steady state iteration process.

The following example will scan the parameter RPM over the range from 19000 to 16000 in five steps. At the end of the steady state calculation, the

system stability will be checked to assure that a stable steady state exists.

```
SS PARAMETER = RPM,  SS START = 19000,  SS STOP = 16000
SS POINTS    = 5, STEADY STATE.
```

If plots of the steady state scan are desired, these plots should be defined using the DISPLAY commands prior to initiating the steady state calculations. Only those plots which have an independent variable different from time will be plotted.

In the following example, the steady state parameter is set to a blank phrase. This is accomplished by placing the SS PARAMETER command phrase at the end of a command line. If it is desired to follow the SS PARAMETER program name with other instructions, then the form: SS PARAMETER = NONE may be used. In either case, this causes a single steady state calculation to occur at the current operating point. The results of this calculation are then loaded into the initial condition vector, XIC. The initial default value of SS PARAMETER is a blank phrase so that single steady state calculations will be performed, unless this parameter is set to a non blank name.

```
SS PARAMETER =
STEADY STATE
XIC-X
```

7. LINEAR ANALYSIS COMMANDS

LINEAR ANALYSIS

This program command causes the calculation of a linearized version of the given nonlinear model at the operating point specified by XIC and then calculates the eigenvalues of this linear approximation. A printout of the following quantities are generated by this command:

1. The state operating point (INITIAL CONDITIONS)
2. The state perturbation size (ERROR CONTROL)
3. The integrator status (INT CONTROL)
4. The rates at the operating point

For continuous systems:

5. The system stability matrix
6. A measure of the linearity of each element of the stability matrix if a nonlinear condition is detected.
7. The system eigenvalues, real and imaginery parts, natural frequencies, and damping ratios.

For discrete systems:

8. Continuous states stability matrix (displays inputs to continuous states)
9. Transition matrix for each sample period (displays inputs to discrete states at each sample period)
10. Total system transition matrix
11. System eigenvalues, real and imaginary parts in both Z and S planes and natural frequencies and damping ratios in the S plane.

EIGENVECTOR

The EIGENVECTOR command is similar to the LINEAR ANALYSIS command. However, in response to this command, the modal matrix comprised of the system eigenvectors is also calculated and printed. This command can only be used with models that contain an optimal controller, due to core requirements.

8. STABILITY MARGIN COMMANDS

STABILITY MARGINS

This program command initiates the calculation of the stability margins for those parameters specified by the SM PARAMETERS command. The maximum and minimum values that each specified parameter can take for stable system operation and the oscillation frequencies that result if either boundary is violated are determined.

SM PARAMETERS

This program command allows up to ten parameters to be specified for stability margin calculations. The command is followed by from one to ten parameter names separated by delimiters. This command destroys all previously stored stability margin parameters.

An example use of these commands is given below:

```
SM PARAMETERS = GK1TC, GK2TC
STABILITY MARGINS
```

These commands cause the stability margins to be calculated for the two parameters, GK1TC and GK2TC.

A summary of stability margins and frequencies is printed along with the nominal system eigenvalues, and the system eigenvalues with each stability margin parameter set equal to zero. If no upper or lower stability margin is located for a particular stability margin parameter, the summary array will contain the number 1111. in those locations for which no margin limit was determined.

The stability margin search is limited to parameter values of the same algebraic sign as the nominal value. Thus, for example, zero is the lowest magnitude that will be considered for the lower stability boundary of a parameter with a positive nominal value.

9. FREQUENCY RESPONSE COMMANDS

<u>TRANSFER</u> <u>FUNCTION</u>
<u>TF</u> <u>INPUT</u>
<u>TF</u> <u>OUTPUT</u>

These program commands are used to initiate the calculation of a frequency response function, between any two specified points in the model. The following command phrases are used to set up the desired transfer function:

TF INPUT = transfer function input variable
TF OUTPUT = transfer function output variable

They are used to specify the input and output points in the system model. These quantities must be set to the desired names before requesting the frequency response calculation. They may be set to any valid state, rate, variable, or parameter name. The command TRANSFER FUNCTION causes the frequency response function to be executed at that point.

The transfer function poles and zeros are printed output. For discrete systems, these roots are given in both the Z plane and S plane.

<u>BODE</u>
<u>NYQUIST</u>
<u>NICHOLS</u>

These program commands specify the format to be used for the frequency response plots. The format must be specified before requesting the TRANSFER FUNCTION analysis. If not specified, the default will be a Bode plot format.

<u>TF</u> <u>AUTO</u> <u>SCALES</u>
<u>TF</u> <u>MANUAL</u> <u>SCALES</u>
<u>FREQ</u> <u>MIN</u>
<u>FREQ</u> <u>MAX</u>

These program commands are used to set the frequency range of the frequency response plots. It can be either automatically determined by the range of eigenvalues or be specified by the following command phrases:

1. FREQ MIN = minimum frequency, r.p.s.
2. FREQ MAX = maximum frequency, r.p.s.

The default condition is for automatic scales.

In the automatic mode, the minimum and maximum frequencies will be one decade below and one decade above the lowest non zero and highest natural frequency. For discrete systems, the upper frequency is bounded by the Nyquist frequency of the system. Frequency points are concentrated around lightly damped natural frequencies to better define these critical areas.

The following example will generate a transfer function from C4 MC to S2 LA with automatic frequency values for the plotted results in a Nichol's chart format.

```
TF INPUT = C4 MC,  TF OUTPUT = S2 LA
NICHOLS, TRANSFER FUNCTION
```

10. ROOT LOCUS COMMANDS

<u>ROOT</u> <u>LOCUS</u>
<u>RL</u> <u>PARAMETER</u>
<u>RL</u> <u>START</u>
<u>RL</u> <u>STOP</u>
<u>RL</u> <u>POINTS</u>

These program commands initiate the calculation of a root locus. The following commands are used to select the parameter and the ranges for the locus.

1. RL PARAMETER = root locus parameter name
2. RL START = initial value of root locus parameter
3. RL STOP = final value of root locus parameter
4. RL POINTS = number of rootings to be made going from RL START to RL STOP

They specify the parameter to scan from the value RL START to RL STOP in RL POINTS steps. The default values of RL PARAMETER, RL START, RL STOP, and RL POINTS are; blank, 0., 1., and 6. respectively.

The root locus parameter, like the steady state parameter, can be either a valid parameter name or a state variable name followed by the phrase IC. This latter usage is meaningful only if the specified state variable has been frozen using the INT CONTROL command. In this way, a root locus can be performed as a function of the operating point value of a frozen state variable.

```
RL PARAMETER = ZO TF,  RL START = 0,  RL STOP = 5,  RL POINTS = 6,
ROOT LOCUS
```

In this example, the root locus parameter ZO TF is scanned from 0 to 5 in six equally spaced steps.

```
RL MANUAL SCALES,  REAL MAX=5,  IMAG MAX=5,  INT CONTROL,  SPEED=0
RL PARAMETER = SPEED,  IC,  RL START = 35,  RL STOP = 45
ROOT LOCUS
```

In this example, manual scales are specified for the root locus plots. The SPEED state variable is then frozen and a root locus is performed on the SPEED operating point.

<u>RL</u> <u>AUTO</u> <u>SCALES</u>
<u>RL</u> <u>MANUAL</u> <u>SCALES</u>
<u>REAL</u> <u>MIN</u>
<u>REAL</u> <u>MAX</u>

IMAG MIN

IMAG MAX

These program commands allow the scales of the root locus plots to be either automatically determined by the range of eigenvalues or to be specified by control commands. The following command definitions are used to set plot scales:

1. REAL MIN = minimum real axis range, r.p.s.
2. REAL MAX = maximum real axis range, r.p.s.
3. IMAG MIN = minimum imaginary axis range, r.p.s.
4. IMAG MAX = maximum imaginary axis range, r.p.s.

The default condition is for automatic scales.

11. EIGENVALUE SENSITIVITY COMMANDS

EIGEN PARAMETER

EIGEN SENSITIVITY

These program commands cause a linear approximation of the given nonlinear model to be generated and then evaluates the sensitivity of the system eigenvalues to a parameter specified by the command phrase EIGEN PARAMETER.

In the following example, the sensitivity of system eigenvalues to the parameter GPITF will be calculated.

EIGEN PARAMETER = GPITF, EIGEN SENSITIVITY

12. FUNCTION SCAN COMMANDS

<u>SCAN1</u>
<u>SCAN2</u>
<u>DEPEN</u>
<u>INDEP1</u>
<u>INDEP2</u>
<u>START1</u>
<u>STOP1</u>
<u>START2</u>
<u>DELTA2</u>
<u>CURVES2</u>

These program commands initiate and control the calculation of general algebraic functions of one or two independent variables. The following definitions are used to specify the control parameters and bounds for the calculation.

1. DEPEN = dependent variable
2. INDEP1 = 1st independent variable
3. INDEP2 = 2nd independent variable
4. START1 = starting point of 1st independent variable
5. STOP1 = stopping point of 1st independent variable
6. START2 = starting point of 2nd independent variable
7. DELTA2 = increment of 2nd independent variable
8. CURVES2 = number of values of 2nd independent variable

These commands specify the dependent and independent variables and scan ranges of these quantities. These quantities must be set to their desired values, before requesting the general algebraic function evaluation. If a single function is requested, i.e., SCAN1, only items 1,2,4, and 5 need be specified.

```
DEPEN = W2 TU,  INDEP1 = EH SH,  INDEP2 = S1 DE2,  START1 = -30
STOP1 = 100,  START2 = 10,  DELTA2 = 20,  CURVES2 = 6
SCAN2
```

In the above example, the quantity W2 TU will be calculated as a function of quantities EN SH and S1 DE2. Six curves will be generated with W2 TU ranging from -30 to 100 and S1 DE2 being stepped from 10 to 20 in 6 steps of 2 each.

13. OPTIMAL CONTROLLER DESIGN COMMANDS

In order to design an optimal controller using the EASY program, it is necessary to specify the inputs and outputs of the optimal controller as part of the system model description. This is accomplished as described in Section II.2.b. Once a model has been generated that contains an optimal controller and the specified input-output connections to the other model components, many different controllers can be designed. These variations are made by varying the operating point or the optimal controller design criteria. The following paragraphs describe how the optimal controller operating point and criteria are specified.

Once an optimal controller has been designed, it may be desired to save that design for further analysis on subsequent analysis runs. Program commands are provided to save the data arrays which specify a particular optimal controller and to read such data on subsequent analysis runs.

O.C. DATA

The O.C. DATA command specifies that the following command phases contain data for one or more of the ten different data arrays related to optimal controllers. The name of each of these arrays and a brief description of its use is given below. For a more complete description of each array and its use, see Section 4.5 of reference 1.

Optimal Controller - Operating Point Specification

YOP - Optimal controller input operating point (set-point array. YOP is an n_s dimensional array, where n_s is the number of inputs to the optimal controller. Default values of zero are provided for this array.

UOP - Optimal controller output operating point (set-point) array. UOP is an n_u dimensional array, where n_u is the number of outputs from the optimal controller. Default values of zero are provided for this array.

Optimal Controller Criteria Specification

Q - Optimal controller criteria weights array. Q is an n_c dimensional array, where n_c is the number of optimal controller criteria variables. Q contains the diagonal elements of the positive semi-definite weighting matrix which gives the importance of the various criteria variables relative to each other and the controller outputs. Off diagonal elements are assumed equal to zero. If the criteria variables are not specified, they are assumed to be the optimal controller inputs. Default values of 1 are provided for this array.

RU - Optimal controller control weights array. RU is an n_u dimensional array, where n_u is the number of optimal controller outputs. RU contains the diagonal elements* of the positive definite matrix which gives the importance of the various controller outputs relative to each other and the criteria variables. Off diagonal elements are assumed equal to zero. Default values of 1 are provided for this array.

CD - System model disturbance covariance array. DC is an n_x dimensional array, where n_x is the order of the system model. DC contains the diagonal elements* of the model disturbance covariance matrix which gives the uncertainty of various model states relative to each other and the sensed quantities. Off diagonal elements are assumed equal to zero. Larger values in CD imply greater uncertainty (less confidence) in the system model accuracy. Default values based on the ERROR vector and the model stability matrix are provided for this array.

CS - Optimal controller inputs disturbance covariance array. CS is an n_s dimensional array, where n_s is the number of inputs to the optimal controller. CS contains the diagonal elements* of the sensed quantity disturbance covariance matrix which gives the uncertainty of various sensed quantities relative to each other and the model states. Off diagonal examples are assumed equal to zero. Larger values in CS imply greater uncertainty (less confidence) in the sensea quantity accuracy. Default values based on the ERROR vector and the model sensor matrix are provided for this array.

Optimal Controller Specification

These inputs are required only for reloading a previously designed optimal controller. Default values of zero are provided for these arrays until nonzero values are calculated via the DESIGN O.C. command.

G - Optimal controller gain array. G is an n_u by n_{rc} dimensional array, where n_u is the number of outputs from the n_{rc} is the order of the optimal controller.

S - Optimal controller sensor array. S is an n_{rc} by n_s dimensional array, where n_{rc} is the order of the optimal controller and n_s is the number of inputs to the optimal controller.

AK - Optimal controller stability matrix array. AK is an n_{rc} by n_{rc} dimensional array where n_{rc} is the order of the optimal controller.

FK - Optimal controller d.c. gain matrix array. FK is an n_u by n_s dimensional array where n_u is the number of outputs from and n_s is the number of inputs to the optimal controller.

Optimal controller array data may be entered in a free field format with each data item separated by a comma or another one of the standard delimiters. Data may be entered along either a row, column or diagonal line of the array. The row and column location is given for only the first element specified. The following input values are loaded in the subsequent row, column, diagonal elements of the array. The letters, C, R, and D signal the start of a new Column, Row, or Diagonal input. They must be followed by the row and column number at which data loading is to start. A column number of 1 <u>must</u> be given for the one dimensional arrays: YOP, UOP, Q, RU, CD, and CS. The letter Z causes all elements of the array to be set to zero. This command may be used to advantage when loading a sparse array.

If the number of data values exceeds either the row or column dimension of the array, the excess values are ignored by the program.

The following example demonstrate the loading of data into the optimal controller arrays.

PROGRAM COMMANDS

```
O.C. DATA
YOP = C (1,1) 553.2, 546, -2.56, 7
```

RESULTS - Assuming YOP is a 4x1 array.

```
      553.2
      546.
YOP = -2.56
      7.00
```

DESIGN O.C.

The DESIGN O.C. command initiates the optimal controller design process. Before issuing this command, the following items should be accomplished:

1. Specify the optimal controller operating point by loading the arrays YOP and UOP.

2. Place the system model at the desired operating point.

3. Specify those optimal controller criteria arrays Q, RU, CD, and CS which you wish to differ from the default values.

The DESIGN O.C. command causes a linear model of the system to be generated and an optimal controller to be designed. The design results are printed and loaded into the optimal controller arrays G, S, AK, and FK. Manual modifications to the optimal controller can be made via the O.C. DATA command.

SAVE O.C.

The SAVE O.C. command causes the optimal controller arrays G, S, AK, and FK to be placed on local file TAPE3 in a format compatible with the O.C. DATA command. This file may be saved as a permanent file or punched as data cards by the appropriate control cards. By including these cards or records in the input data for subsequent analysis runs, it is possible to perform further analyses on a previously calculated optimal controller. Such optimal controller data could be used in conjunction with the O.C. ANALYSIS command to the Model Generation Program. As described in Section II.2.b, the O.C. ANALYSIS command allows analyses to be performed on a previously designed optimal controller with less computer central memory than is required to perform the optimal controller design.

14. WARNING MESSAGES

One or more of the following warning messages will occur if the program encounters difficulty in interpreting analysis instructions or performing an analysis. These messages will be preceded by:

*** WARNING ***.

The symbols xxx, zzz, or nnn are used to indicate phrases from the analysis description that are included as part of the warning message. The following messages are listed in alphabetical order:

1. A VALID PARAMETER NAME MUST PRECEDE THE NUMERIC VALUE nnn

This message indicates that a valid parameter name was not identified preceding the numeric value nnn. Check for missing delimiters or misspelled parameter name.

2. ALGEBRAIC LOOP WITH GAIN OF nnn EXISTS BETWEEN INPUT AND OUTPUT THIS TRANSFER FUNCTION CAN NOT BE DETERMINED.

See Appendix M for a description of this limitation to the transfer function analysis method.

3. ALL ROOTS CANCELED. THIS CASE WILL BE SKIPPED

This indicates TF output is not conected to TFD input. Check model, TF input, and TF output specifications.

4. nn IS NOT A VALID SUBSCRIPT

Subscripts must be numeric.

5. xxx IS NOT A VALID TABLE NAME

Check spelling of table name.

6. xxx IS NOT A VALID TABLE NAME FOR THIS MODEL. DATA WILL BE IGNORED

Check spelling of table name.

7. CAN'T FIND GREATEST COMMON DIVISOR FOR THE FOLLOWING SAMPLE RATES

Check sample period values.

8. CAN'T FIND LEAST COMMON MULTIPLE FOR THE FOLLOWING SAMPLE RATES

Check sample period values.

9. CAN'T IDENTIFY xxx AS A VALID EIGENVALUE SENSITIVITY PARAMETER

Check spelling of eigenvalue sensitivity parameter or for missing delimiters.

10. CAN'T IDENTIFY xxx AS A VALID PRINT VARIABLE

Check spelling of xxx or for missing delimiters.

11. CAN'T IDENTIFY xxx AS A VALID ROOT LOCUS

Check spelling of xxx or for missing delimiters.

12. CAN'T IDENTIFY xxx AS A VALID SCAN PARAMETER

Check spelling of xxx or for missing delimiters.

13. CAN'T IDENTIFY xxx AS A VALID STABILITY MARGIN PARAMETER

Check spelling of xxx or for missing delimiters.

14. CAN'T IDENTIFY xxx AS A VALID STEADY STATE PARAMETER

Check spelling of xxx or for missing delimiters.

15. CAN'T IDENTIFY xxx AS A VALID TRANSFER FUNCTION INPUT (OUTPUT) PARAMETER

Check spelling of xxx or for missing delimiters.

16. xxx CAN'T BE SET EQUAL TO zzz. VALUE MUST BE NUMERIC

Check for missing numeric value or delimiters.

17. CAN'T IDENTIFY xxx VALUE WILL BE IGNORED

This will result in not setting the quantity intended by xxx to its new value. Check for spelling of xxx or for missing delimiters.

18. CAN'T INTEPRET xxx

The phrase xxx cannot be recognized as a valid program command, program name, or program value. Check spelling of xxx or for missing delimiters.

19. CAN'T LOAD CRITERIA ARRAYS WHEN IN ANALYSIS ONLY MODE

The O.C. ANALYSIS command was issued to the Model Generation program when it created the system model. Therefore, an optimal control design, which used this criteria arrays, cannot be performed.

20. INVALID SUBSCRIPT DETECTED

Subscript outside valid range for this array.

21. SUBSCRIPT VALUES nn OR nn ARE TOO LARGE FOR xxx

Subscripts outside allowable range.

22. WORK SPACE WAS NOT PROVIDED IN MODEL FOR OPTIMAL CONTROLLER DESIGN OR EIGENVECTOR CALC.

An optimal controller must be specified in model description in order to have work storage for optimal control design of eigenvector calculation.

23. nnn EXCEEDS THE ALLOWABLE INDEX RANGE FOR xxx THIS QUANTITY WILL NOT BE DEFINED

The number nnn was outside the allowable range of states, rates, variables, or parameters. Therefore, the name xxx cannot be assigned as a name for the nnnth state, rate, variable or parameter.

24. nn IS OUTSIDE ALLOWABLE INDEX RANGE. zzz WILL NOT BE DEFINED

Index number nn must be between 1 and number of states, variables, or parameters, (whichever is applicable).

25. FAILED TO CONVERGE TO ZERO PHASE

The search procedure described in Appendix M failed to converge to zero phase. The stability margin for the indicated parameter cannot be determined by this method.

26. MORE THAN 10 UNIQUE SAMPLE RATES LOCATED

Only 10 different sample rates allowed.

27. NO SAMPLING PERIODS ARE GIVEN

Sampling period parameters TAU xxx could not be located. These names can not be redefined.

28. NOMINAL SYSTEM UNSTABLE

The nominal system is unstable. The stability margins of the specified parameters will be calculated, but these bounds will be "non-critical" bounds since the nominal system is unstable. See Section 4.4.4 of reference 1 for a discussion of critical and noncritical stability boundaries.

29. NON-ALPHA NAME ON THIS CARD --- xxx. WILL IGNORE THIS CARD

The table inputs routine expected an alphanumeric table name but encountered a numeric value on the data card printed. Check the sequence and number of tabular data cards to assure that they match those required by the model's tables and table input formats. See Section III.1.b for correct formats.

30. NON-NUMERIC DATA ON THIS CARD --- xx. WILL READ NEXT TABLE

The table input routine expected a numeric value but encountered an alphanumeric name on the data card printed. Check that the sequence and number of tabular data cards matches the model's tables and table input formats. See Section III.1.b for correct formats.

31. nnn PRIMARY AND xxx SECONDARY INDEPENDENT VARIABLE POINTS EXCEEDS THE zzz WORD STORAGE LIMIT FOR THE FOLLOWING TABLE. SOME DATA WILL BE LOST

See Section II.2 for a discussion on how to set the maximum number of data points allowed for each table.

32. SIMULATION WILL NOT BE RUN DUE TO FAILURE TO REACH VALID STEADY STATE

A failure of the steady state analysis followed by a request to transfer X into XIC causes an interlock to be set which will prevent a simulation run from beginning from an erroneous initial condition.

33. WORK SPACE WAS NOT PROVIDED IN MODEL FOR OPTIMAL CONTROLLER DESIGN

Either no optimal controller was specified to the Model Generation Program or the O.C. ANALYSIS mode was indicated. In either case, only analyses and not DESIGN O.C. can be performed with this model.

34. *** WARNING *** MATRIX IS SINGULAR *** INITIAL SYSTEM IS NOT DIAGONALIZABLE

This message is generated in the system reduction program and is the result of multiple eigenvalues with a single eigenvector. This means that the system is not able to be diagonalized and that a Jordan type reduction is required. Processing is stopped and reduction is not completed. This message can arrise either in the reduction of the initial model equations or in the reduction of the controller.

35. *** WARNING *** QR FAILED TO CONVERGE IN XX STEPS

This message generated in the system reduction program is the result of the extremely rare event of the eigenvalue calculation failure.

36. ** DUE TO xxx UNSTABLE EIGENVALUES. SYSTEM REDUCTION TO xxx IS IMPOSSIBLE

This message generated in the system reduction program is the result of the number of unstable eigenvalues in the system to be reduced being greater than the requested order for the reduced system. This message can arise either in the reduction of the initial system or in the reduction of the controller.

37. ** CONTROL WEIGHTING NOT POSITIVE DEFINITE

This message generated in the calculation of the optimal feedback matrix in the result of loss of significance in the calculation of the control weighting matrix. Since the default check is made, this is a rare event.

38. **... QR ALGORITHM FAILED TO CONVERGE
**... SYSTEM MAY BE UNSTABILIZABLE

This message generated in the calculation of the optimal feedback matrix is the result of the QR algorithm failure and is a rare event.

39. **... SPECTRAL FACTORIZATION OF EIGENVALUES NOT OBTAINED
**... SYSTEM MAY BE UNSTABILIZABLE

This message generated in the calculation of the optimal feedback matrix is the result of an eigenvalue with a zero real part preventing spectral factorization. It is the result normally of an uncontrollable mode with an eigenvalue with a zero or very small real part.

40. **... MATRIX IS SINGULAR

**... SYSTEM PLUS ADJOINT EQUATIONS NOT DIAGONALIZABLE OR SYSTEM IS UNSTABILIZABLE

This message generated in the calculation of the optimal feedback matrix is the result of the set of pseudo eigenvectors calculated for the partitioned eigenvalues being singular in the top block. This condition normally means that an unstable, uncontrollable mode existed in the original system. Another, but rare possibility is that due to multiple eigenvalues, the system plus adjoint equations was not diagonalizable.

41. **... OR FAILED TO CONVERGE

**... SYSTEM MAY BE UNOBSERVABLE

This message is generated during the calculation of the Kalman filter and is the result of the QR algorithm failure and is a rare event.

42. **... SPECTRAL FACTORIZATION OF EIGENVALUES NOT OBTAINED

**... SYSTEM MAY BE UNOBSERVABLE

This message is generated during the calculation of the Kalman filter and is the result of an eigenvalue with zero real part preventing spectral factorization. It is normally the result of an unobservable mode with an eigenvalue with zero or very small real part.

43. **.. MATRIX IS SINGULAR

**.. SYSTEM MAY BE UNOBSERVABLE

This message is generated during the calculation of the Kalman filter and is normally the result of an unstable unobservable mode. Like the case in the gain matrix calculation (Section 4.6.30 of reference 1), it can rarely be the result of the system and adjoint equations being undiagonalizable.

44. **... QR ALGORITHM FAILRED TO CONVERGE

This message occurs when during a simple eigenvalue calculation, convergence was not obtained. This is a rare event.

45. **... SYSTEM HAS SINGULAR ALGEBRAIC LOOP

This message generated during the adjustment of the controller is the result of cancellation in algebraic feedforward and feedback loops. It can normally be corrected by the use of an alternative adjustment method.

15. RENAMING MODEL INPUTS AND OUTPUTS

For some applications, it may be desirable to rename the parameters, states, rates, and variables created by EASY5 standard components. This can be done by the following analysis program commands:

<u>DEFINE PARAMETERS</u>
<u>DEFINE STATES</u>
<u>DEFINE RATES</u>
<u>DEFINE VARIABLES</u>

Each command is followed by pairs of names. The first name is the EASY5 standard component name. The second name is the desired new name. For example, the outputs of the lag component LA may be changed to AILERON, and the lag gain and time constant may be changed to KSERVO and TSERVO.

DEFINE STATES = S2 LA = AILERON
DEFINE PARAMETERS = GAILA = KSERVO, TC LA = TSERVO

Once a quantity has been redefined, all references to that quantity in analysis program commands must utilize the new name. The subroutine EQMO, which is prepared by the EASY5 Model Generation Program, will still refer to all quantities by their original EASY5 generated names.

16. COMPUTING TYPE ZERO TRANSFER FUNCTIONS WITH EASY

A continuous dynamical system (with prescribed input and output quantities) has a Type Zero transfer function if either:

1. A change quantity in the input has an immediate change in the output quantity

or equivalently:

2. The order of the numerator of the transfer function is the same as the order of the denominator

The method currently used by the EASY Dynamic Analysis Program is unable to compute transfer functions of systems of Type Zero. This will be remedied in the future, but the following provides an interim method:

A. In the model description file:

1. Add a new LA standard component. We will name this component LATF but you may use any unused component identifier.
2. Connect the output of the new LA component to the original system input quantity.

B. In the system analysis file:

1. Set the parameters for the new LA component as
   ZOLATF = 1   ZILATF = 0   POLATF = -1.0E28
2. Change the TF INPUT quantity from the original quantity to S1 LATF

C. Submit job using new model description and analysis files.

D. The results of the TRANSFER FUNCTION analysis will provide:

1. The zeros and poles of the original system plus a pole at 10^{28} radians per second. This extra pole should be ignored.

2. The frequency response will be the correct frequency response for the original system up to frequencies above 10^{20} radians per second.

These high frequency values can be suppressed from the lineprinter output and the graphs by using the TF MANUAL SCALES option.

SECTION IV

STANDARD COMPONENTS AND EASIEST SUBROUTINES

This section describes the EASIEST standard components available for system modeling that were designed from the SAFEST computer program. Other components that may be used by the analyst in conjunction with the EASIEST routines are described in Appendix K.

1. Standard Components

The following is a list of the EASIEST standard components:

NAME	DESCRIPTION
AB	Attached body (survival kit)
AE	Airplane
AG	Atmospheric properties
AM	Aeromedical
AP	Aerodynamic plate
AS	Seat aerodynamics
CE	Crewperson
CS	Airplane control surfaces
CT	Catapult
DR	DART
GP	Simple parachute mortar and restraints
LI	Parachute lines
MP	Parachute mortar and restraints
PC	parachute
RL	Rails
RS	Restraints
SE	Seat equations of motion
SL	Sled
SP	STAPAC
SR	Sustainer rocket
WB	Weight and balance

This section gives an explanation of each of the aforementioned ejection seat components. These descriptions are intended to assist the user in utilizing them to model escape systems. Input/output tables and descriptive figures for each of these components are presented in alphabetical order in Appendix D, and should be thoroughly examined before modeling an ejection system.

A source listing of the EASIEST components and associated subroutines are presented in Appendices G and H. These listings have been thoroughly commented to provide additional information on how the algorithms were coded and to assist in solving special case errors.

STANDARD COMPONENT AB

This component is simply the equations of motion for a point mass. It was designed to model a survival kit attached to the crew member, but can be used to simulate any object that might be attached to the escape system. Component restraints (RS) is used to restrain AB to its parent object. The input/output list for this component is given in Appdendix D. Inputs include the forces and torques that act on the point mass, as well as it's inertial properties.

STANDARD COMPONENT AE

This component models the EASIEST airplane. The airplane is internally trimmed by the STEADY STATE command to the airspeed and altitude specified by the user. Control surface and thrust commands that maneuver the airplane after trim are interpreted as being an addition to the settings required for trim. Additional inputs include the forces and torques from the DART, rails, and catapult components. An example of a model that uses component AE is given in Appendix N. Additional airplane information is presented in Section IV.3.

Component AE was written to use existing SAFEST aerodynamic coefficient tables and table look-up routines with the exception that coefficient

AD-A096 597 BOEING MILITARY AIRPLANE CO SEATTLE WA F/G 1/3
ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM. VOLUME --ETC(U)
SEP 80 C L WEST, B R UMMEL, R F YURCZYK F33615-79-C-3407
UNCLASSIFIED AFWAL-TR-80-3014-VOL-1 NL

2 OF 8

AD A
096597

input data has been reorganized so they contain the coefficients in the following order:

NR LOCATION	COEFFICIENT NAME	DESCRIPTION
1	CZO	Z axis bias coefficient
2	CZAD	Variation of CZO with alpha dot
3	CZQ	Variation of CZO with pitch rate
4	CZDE	Variation of CZO with elevator position
5	CZDA	Variation of CZO with aileron position
6	CXO	X axis bias coefficient
7	CXDA	Variation of CXO with aileron position
8	CMO	Pitching moment bias coefficient
9	CMAD	Variation of CMO with alpha dot
10	CZQ	Variation of CMO with pitch rate
11	CMDE	Variation of CMO with elevator position
12	CMDA	Variation of CMO with aileron position
13	CYB	Variation of CY with beta
14	CYP	Variation of CY with roll rate
15	CYR	Variation of CY with yaw rate
16	CYDR	Variation of CY with rudder position
17	CYDA	Variation of CY with aileron position
18	CLB	Variation of Cl with beta
19	CLP	Variation of Cl with roll rate
20	CLR	Variation of Cl with yaw rate
21	CLDR	Variation of Cl with rudder position
22	CLDA	Variation of Cl with aileron position
23	CNB	Variation of Cn with beta
24	CNP	Variation of Cn with roll rate
25	CNR	Variation of Cn with yaw rate
26	CNDR	Variation of Cn with rudder position
27	CNDA	Variation of Cn with aileron position

A listing of the F4E airplane maneuvering coefficients modified to be used with EASIEST is shown in Appendix J.

Component CS (airplane control surfaces) can be used to maneuver the airplane. The method employed to do this is described in this section under the heading STANDARD COMPONENT CS. This component is also included in the example presented in Appendix N.

STANDARD COMPONENT AG

Component AG calculates the atmospheric density and the speed of sound, while supplying the wind velocity to the model. It should be the first component specified in the Model generation Program input file, and must be included in all EASIEST models.

Note that variables H, BP, and TE must be initialized if a non-standard atmosphere is to be used with the model. Setting variable BP to zero, which is it's default, establishes a standard atmosphere. The wind velocity input vector provides the capability to model an ejection system where adverse winds (i.e., storm cells, turbulence, down drafts, etc.) could be a factor in an ejection seat design. This feature may be valuable when using the EASIEST program to investigate an aircraft accident.

During initialization (CALC XIC command), component AG establishes the atmospheric properties from the input parameters. Subsequent passes through the model updates the wind vector. If a standard component needs atmospheric data, it is acquired by a call to subroutine ATMOS, which refers to the ENTRY ATMOS statement in component AG.

STANDARD COMPONENT AM

This component acts essentially as the interface between program Aeromed, the aeromedical post processer, and either component SE (seat equations of motion) or CE (crewperson). The routine writes onto TAPE 7 the aeromedical parameters and variables required by Aeromed. This process is initiated by

a flag that is an input into the component. No more than 4000 variable sets can be written to this tape at a time interval no less than 0.001 seconds, or the integrator report interval, TINC, whichever is the largest. (See Section III.4 for an explanation of TINC.)

Components CE and SE both calculate the aeromedical variables, and either one can be used to drive the aeromedical inputs in this component. Note that most of the required parameter inputs have specific defaults, which can be adjusted by the user if necessary.

STANDARD COMPONENT AP

This EASY module calculates the seat body axis force and torque components acting on the ejection seat from an attached object, such as an airfoil device or inflatable afterbody designed to augment the stability of the ejection seat. Appendix D presents its input/output lists. Inputs include the tables that define the x-axis and the z-axis force coefficients, the plate centroid in the seat coordinate system, and the airplane z-axis position at the point where the plate centroid enters the windstream. The plate centroid acts as the origin of the plate coordinate system, and the plate can be rotated about this point with respect to the seat. Figure 22 provides an input/output overview for this component.

STANDARD COMPONENT AS

Component AS determines the aerodynamic forces and torques that are exerted on the seat. It employs the same coefficient input data and table look-up routines as the SAFEST program. The input/output information is contained in Appdendix D. Inputs include emergence coefficients, the yaw, pitch, and roll damping derivatives, and a table that defines the exposed area of the seat as a function of the exposed length during emergence. Figure 23 presents a diagram that helps to explain the function of component AS.

Both the rocket on and rocket off aerodynamic coefficient tables are available at any given time to accommodate the situation where two ejection seats are being modeled, one of which has its rocket on, the other off. Each of these coefficient tables are hard coded into this component, and contain the six basic aerodynamic coefficients: the three body axis force coefficients (CX, CY, CZ), and the three body axis torque coefficients (Cl, Cm, Cn).

STANDARD COMPONENT CE

This EASIEST standard component computes the aerodynamic forces and torques acting on the percentile crewperson that is specified in the CE input data. These forces and torques are then summed with the other forces and torques acting on him (parachute lines, seat restraints, etc.) to determine the linear and angular rates to be used by the integrator. The input/output listings are presented in Appendix D.

Note that the moments and products of inertia for the crewmember are required inputs. The values for these parameters should reflect the inertial properties of a seated crewmember whose percentile is approximately the same as that specified in the input data. At seat/crewmember separation, new moment and product of inertia vectors are calculated via a table look-up on data hard coded into the component, with the independent variable being the crew member percentile. The aerodynamic reference area and length are also determined by this table look-up, as is the crewmember weight. The weight of the crewmember's clothing and equipment is a separate parameter input.

STANDARD COMPONENT CS

This component can be employed to move the rudder, elevator, and ailerons of the airplane component (AE). All three control surfaces may be moved simultaneously or individually according to the input parameters specified by the user. These parameters include the simulation time after which the control surface rates are calculated, the commanded position, and a time

constant that is employed by a first order lag function to determine the rates. The input/output data is given in Appendix D.

COMPONENT CT

Component CT determines the forces and moments acting on the seat and airplane from a closed tube catapult. The states in this module include the internal friction energy, heat loss, catapult work, and the propellant web consumed. These states are used to calculate the internal temperature of the catapult, from which the pressure is calculated by using the equation of state with the chamber volume and the mass of the burned propellant. The force can then be calculated from the geometry of the catapult pressure chamber.

The input/output parameters are shown in Addendix D, and include the flag for catapult ignition, the unloaded catapult length, and the catapult propellant consumption table. Figure 24 presents an overview of some of the required inputs, and should be helpful in visualizing the geometry and operation of the catapult. Note that input TDE is available as the time interval over which the catapult force decays to zero after stripoff. This decay period should prevent the variable step integrators from the difficulties associated with sudden rate changes.

STANDARD COMPONENT DR

This standard component simulates the "DART" stabilizing device that can be used by an ejection seat to correct for adverse pitch and roll induced by aerodynamic torques and the offset caused by improper allignment of the seat center of gravity with the sustainer rocket thrust vector. It is not effective in providing corrective torques about the yaw axis.

The DART is a simple device which consists of a line that is connected at one end to the airplane, and at the other end to a bridle attached to the bottom of the seat. This line passes through a braking device, whose force is calculated from a table that is an input into the component. This

table, as well as the other input/outputs, are explained in Appendix D. In addition, Figure 25 provides a descriptive diagram for this component.

STANDARD COMPONENT GP

This standard component is a simplified version of component MP (parachute mortar), in that a table look-up is used to find the mortar force as a function of time, instead of the equation of state method employed by MP. The input/output list is given in Appendix D. Due to a configuration where the mortar force vector may not pass through the parachute center of gravity, inputs for both the position of the parachute attachment point and the seat deployment impulse arm are required. In this situation, the force imparted from the gun to the pack is assumed to act parallel to that of the gun impulse vector.

Component GP also has the task of restraining the parachute to the seat prior to mortar initiation. When the mortar is fired, and the chute is propelled away from the seat, the restraint logic prevents the parachute from moving perpendicular to the mortar impulse vector until the mortar reaches stripoff. Mortar stripoff is defined as the time the mortar force reaches zero, which is set in the mortar force input table. When the mortar reaches stripoff, the forces and torques acting on the seat and parachute calculated by the restraint logic are set to zero. However, these forces and torques may be gradually reduced to zero over a time period, defined by input DCE, if desired by the user. This capability was included in the component to prevent the variable step integrators from having difficulty with a sudden rate change.

STANDARD COMPONENT LI

This component calculates the forces and torques that are imparted from a loaded parachute line onto an object that is being decelerated by a parachute. The input/output list is shown in Appendix D. The inputs include the states from both the decelerated object and parachute. Additional

inputs define the bridle configuration and the parachute line characteristics.

The subroutines that are used by component LI include LILOAD, which calculates the line load; LIBRIDL, a routine that determines the force application point; and LILINE, an algorithm that calculates various line parameters. LILOAD is the line model described in reference 2. Subroutine LIBRIDL can accommodate bridles that have one through four attachment points. If there is only one attachment point, the force application point is set equal to the position of attachment point one, and the input defining the bridle apex, namely APX, should be set to zero. Variables calculated in LILINE include the parachute line length, defined as the distance from the stretched canopy center of gravity to the force application point.

STANDARD COMPONENT MP

This module is the EASIEST parachute mortar model, and closely resembles components CT (catapult) and RS (restraints), in that logic similar to that in CT is employed to calculate the force generated by a closed tube telescoping catapult, while the RS logic is used to maintain the parachute's position on the seat until the mortar is initiated. From mortar initiation until stripoff, the restraint logic maintains the parachute on a path that is defined by its seat attachment point and the mortar force vector.

Appendix D gives the inputs and outputs for this component. Inputs include parameters that define the characteristics of the mortar's performance and the spring and damping constants for the restraints. Input TDE is the time interval over which the mortar and restraint forces decay to zero. This input was included to prevent the variable step integrators from having difficulties with sudden rate changes.

STANDARD COMPONENT PC

This module is the EASIEST parachute model. It is capable of modeling either a drag chute or a recovery chute by setting the input data to

correspond to the type of parachute desired. The inputs include variables from components LI and the parachute mortar (GP or MP), as indicated in the input/output descriptions in Appendix D. Additional information concerning the input data is presented in Figure 26.

This component calculates rates for both the parachute pack, defined as the parachute container and the canopy/lines contained within it, and the canopy. Prior to linestretch, the mass of the canopy is set at one pound and driven to the calculated stretched canopy center of gravity by a spring, whose characteristics are defined by input parameters CSP and DPG. After linestretch, the parachute container separates from the canopy, with only the force of gravity acting on it. However, since the container has a coordinate sytem attached to it, its rotation must be stopped to prevent the Euler angle singularity, an occurrence which reduces execution efficiency when using the variable step integrators. This is accomplished with input DPG, a user defined vector which induces a braking torque about all three axes of the pack's coordinate system. Another input, TEM, is the time duration over which the aerodynamic forces are factored during parachute emergence into the windstream. It also performs a similar function when the lines are severed, ensuring variable step integrator efficiency.

This algorithm is separated into three distinct phases. Phase one is concerned with the parachute dynamics prior to parachute launch. Forces acting on the parachute include the mortar and the restraints. Forces acting on the canopy, which is treated as a separated object, are the spring forces that maintain its position in the pack. Phase two models the parachute from launch to linestretch. Forces that act on the pack include the parachute stripout force and the aerodynamic forces. Forces that are exerted on the canopy are the spring forces that drive the canopy to its center of gravity position along the parachute lines. The center of gravity position is passed to this component from component LI (parachute lines). Phase three takes into account the forces that act on the canopy after linestretch, which include the aerodynamic and line forces, as well as the mass acquisition force as the parachute inflates.

STANDARD COMPONENT RL

This standard component determines the forces and moments that act on the vehicle and the seat while the slider blocks are in contact with the rails. The resulting forces and moments acting on the seat and the vehicle are due to rail elasticity and rail to slider block friction forces. The input/output table is given in Appendix D. Note that states from components SE and the vehicle (AE or SL) are required inputs, and must be accounted for by the component hookups in the Model Generation Program input data. Other inputs include the slider block friction coefficient, and the ejection direction flag. Figure 27 provides an additional explanation for some of the inputs, and helps to explain the rail/slider block geometry.

STANDARD COMPONENT RS

This EASIEST component is the module which restrains one object to another, such as the crewmember to the seat. The input and output data is given in Appendix D. The nomenclature for this component defines the parent body as that object in whose coordinate system the attachment point is defined. The second object is referred to as the attached body. The inputs to this component include the attachment point where the attached body is constrained. The two bodies are held in the relative position defined by the input data by a set of springs which exert both torques and forces on the constrained bodies. The bodies are held together until a switch is set by the sequencer, which is described in Section IV.3.

STANDARD COMPONENT SE

This component sums the forces and torques that act on the seat, and then determines the seat body axis angular and linear rates. The composite seat inertial properties are fed to this component from component WB (weight and balance) if an object is pinned to the seat, as in the case of the sustainer rocket (SR). Otherwise, the inertial properties are inputed directly into the component. Note that the equations of motion were

written so that the linear states apply to the seat reference point rather than the seat center of gravity.

The input/output variables and parameters are given in Appendix D. All pyrotechnic devices, such as the catapult, should have their forces and torques feed into SE via the ports labeled F1 and T1. The forces and torques sent to this component from non-pyrotechnic sources, such as the aerodynamics, should use ports F2 and T2. This constraint is to help the user to organize the inputs into component SE.

STANDARD COMPONENT SL

Component SL is the EASIEST sled model. The linear velocity and position vectors should be initialized in the Analysis Program input data by the INITIAL CONDITIONS command. The angular velocity vector must be initialized to zero, as explained in Section IV.3. Note that the names of the SL states have the same names as those of the airplane, simplifying the process of interchanging the two vehicles in a model file. Appendix D gives a list of the input/output information. Note that the velocity vectors are defined with respect to the sled body axis.

STANDARD COMPONENT SP

This component simulates the STAPAC ejection seat stability device. It consists of a vernier rocket motor connected to a single-degree-of-freedom gyroscope. It can be mounted on the ejection seat to provide a correcting torque for either an adverse yaw, pitch, or roll.

Appendix D supplies the input/output names assigned to this component, while Figure 28 explains the coordinate systems attached to the rocket and the gyroscope. The Euler angles that define the orientation of the rocket and the gyroscope coordinate systems in the seat reference frame are states. Consequently, they must be initialized in the analysis file. Proper initialization can model either a yaw, pitch, or roll STAPAC. Once the gyroscope wheel is spun up and the gimbal uncaged, the seat body axis

angular velocities are projected onto the gimbal axis. If an angular velocity component exists on the input axis of the gyroscope, as shown in Figure 29, the gyro processes, rotating the vernier rock to provide a correcting torque. The forces and torques generated by this rocket are then passed to component SE (seat equations of motion).

Figure 29 provides additional information on the inputs to this component. It explains the biasing effect of the gimbal spring, and what is meant by the thrustline offset. In addition, input TSU specifies a time duration over which the gyroscope wheel accelerates to its uncaged angular velocity. This prevents the variable step integrators from encountering an extreme rate change.

STANDARD COMPONENT SR

The purpose of this module is to calculate the forces and torques that act on the ejection seat from the sustainer rocket. In addition, the inertial properties of the rocket propellant grain are calculated as the rocket burns, and made available to component WB (weight and balance) for the composite seat weight and balance calculation.

Appendix D contains a list of SR input/output descriptions. Figure 30 presents a pictorial explanation of some of these inputs and variables. As shown in the figure, the rocket has a coordinate system attached to the propellant grain center of gravity. In addition, the rocket nozzle has its own corrdinate system, with the thrust vector acting along the negative direction of its z-axis. The location of the origin of the propellant grain is with respect to the seat coordinate system, as are its Euler angles. The location of the rocket nozzle's origin and Euler angles are defined with respect to the propellant grain coordinate system. Because the propellant weight is a state, it must be initialized in the analysis file.

During initialization, the specific impulse of the rocket and the initial propellant moments of inertia are calculated. Once the rocket is switched on by the sequencer, the force generated by the rocket is determined by a

table look-up, the propellant consumption rate is calculated, and the moments and products of inertia of the propellant are updated and rotated into the seat coordinate system.

An additional capability of this module includes utilizing it to model an ejection seat with a "thrust vector control" sustainer rocket. This is demonstrated in the model that is presented in Appendix N.

STANDARD COMPONENT WB

This EASIEST component determines the composite center of gravity and inertial properties of the ejection seat. The sustainer rocket propellant is included in this calculation, but ejection seat components which utilize springs to couple themselves to the seat are excluded. This component can accommodate up to three attached bodies.

The input/output information for WB is given in Appendix D. The inputs include the number of attached bodies, the seat body axis position vector of the basic seat center of gravity, the basic seat moments and products of inertia about the seat center of gravity, and the basic seat weight. In addition, the seat system location of each attached body center of gravity is a required input, along with its weight, and the moments and products of inertia rotated into the seat system. The outputs include the following composite seat properties:

a. Weight
b. Center of gravity in the seat body axis system
c. Moments of inertia about the seat center of gravity
d. Products of inertia about the seat center of gravity

These outputs are passed to component SE to be utilized by the seat equations of motion.

2. SUBROUTINES

The EASIEST subroutines (not standard components) listed in Appendix H that are utilized by the EASIEST standard components are available to the analyst for system modeling, and can be used with the FORTRAN STATEMENTS command. Additional subroutines, whose listings are available in Volume II, Section III, of this document, can also be used in system modeling.

3. MODELING WITH THE EASIEST COMPONENTS

This section covers modeling requirements and methods which must be satisfied when an analyst models an escape system with the EASIEST components. It also will help to explain how to resolve certain problems that may be encountered.

Any of the EASIEST components may be employed as often as required in system modeling. However, component AG (atmospheric properties) must be included in all EASIEST models, since it controls a common statement variable used by some EASIEST components, and supplies atmospheric information to PC (parachute), AS (seat aerodynamics), CE (crewperson), and AE (airplane).

A specific sequence of analysis commands should be followed to properly define input parameters and to initialize the model. The analysis file for the examples in Section VI and Appendix N demonstrate this procedure, and it is listed as follows:

(1) TABLE - allows for the input of a required table.

(2) PARAMETER VALUES - precedes the defining of parameter values.

(3) INITIAL CONDITIONS - permits the initialization of state variables (seat velocity, for example).

(4) CALC XIC - allows for the calculation of variables derived from input parameters (the sustainer rocket's specific impulse, for

example). Parameters not defined after the PARAMETER VALUES command are set equal to their default values.

(5) INT CONTROLS - freeze the required states prior to the issuance of the STEADY STATE command.

(6) STEADY STATE - drives all objects attached to the seat by the restraint components to their attachment position, and determines their velocities.

(7) XIC-X - transfers the states calculated by the steady state solver into the initial conditions vector.

(8) ALL STATES or INT CONTROLS - specifies which states will be used by the subsequent analysis commands.

(9) Desired analysis commands (SIMULATE, LINEAR ANALYSIS, etc.).

A trim scheme has been devised to initialize the states of the physical objects attached to the ejection seat by the restraint components, such as the crewperson and the parachutes. If the sled or airplane is used in the component, the only states that need to be initialized are the vehicle's linear velocity, angular position, and the linear position vectors. (Note: The angular velocity vector of the vehicle must be set to zero, since the steady state scheme cannot accommodate non-zero angular velocities.) After the CALC XIC command is given, all of the vehicle's states are then frozen by the INT CONTROLS command. Model states that are not directly associated with the dynamics of physical objects must also be frozen. These include the states associated with the catapult (CT), mortar (MP), parachute lines (LI), sustainer rocket (SR), and STAPAC (SP). If any of these states are not frozen, the EASY steady state solver will not be able to solve for a steady state, and the command will terminate.

The user must be aware that the STEADY STATE command can calculate an undesired steady state, with the seat driven to an attitude where the plane formed by the slider blocks is perpendicular to the rails. An inverted steady state is also possible. This situation can easily be avoided by initializing the states of the seat as near to their steady state operating point as possible.

Another method to assist the steady state solver is to set the value of parameter SW in component AG to 0.0 before issuing the STEADY STATE command. This prevents the parent objects in the model from "seeing" the forces and torques applied to them by the restraint components. For example, the seat component (SE) will receive rail and catapult forces and torques, but will not receive the forces and torques from the components which restrain the crewperson and the parachutes to it. Likewise, the crewperson will receive forces and torques from the restraints which hold him in the seat, but will not see any forces or torques from anything attached to him. Once a steady state has been calculated with SW AG set to zero, SW AG must be redefined to a value of 1.0, an XIC-X command given, and then the STEADY STATE command repeated. This scheme has been included in the model only as an additional capability, and as a rule it does not have to be implemented.

If an analyst desires to determine the steady state of a seat in a model where there is no vehicle (i.e., the seat is unsupported), then the user must perform the following tasks within the previously described command sequence:

(1) Freeze all of the states in component SE.

(2) Define SW AG to be equal to zero. (Not required if there are no objects attached to the seat.)

(3) Set TM SE equal to the desired earth frame linear trim velocity.

(4) Issue the STEADY STATE command.

(5) Redefine SW AG to be equal to one. (Not required if there are no objects attached to the seat.)

The parameter SW AG, when set to 0.0, has the additional capability of setting the acceleration of gravity to zero throughout the model. If the acceleration of gravity is not set to zero before issuing the STEADY STATE command when the seat is unsupported, the restraint springs would have to load up to resist the acceleration of gravity. After unfreezing the seat states, the model would no longer be at a steady state operating point.

The implementation of the airplane component requires a slightly different procedure. The basic sequence of simulation commands outlined earlier in this section should be adhered to; however, prior to the STEADY STATE command, the only airplane states that need to be frozen are EAPAE(1), XAPAE(1), and XAPAE(2). In addition, the states associated with the control surface component (CS) must be frozen if it is employed in the model. The earth system trim velocity and altitude are required inputs into component AE, and should be set to the desired values. Appendix N contains an example of an EASY model that employs the AE component. The aforementioned method of assisting the EASY steady state solver with SW AG is also demonstrated in this example.

When component AE is included in a model, the airplane aerodynamic coefficients must be made available to it. The procedure used to submit an EASIEST run that includes component AE is explained in Section V. An example of a set of coefficients formatted for component AE is given in Appendix J.

When employing any of the restraint components (namely, RL, RS, GP, or MP) in a model, the spring and damping constants associated with them must be defined in such a manner as to set the system's natural frequencies below approximately 1000, and the damping ratios between 0.6 and 0.9. The recommended approach to do this is to first set the angular and linear spring terms according to the magnitude of the attached object's inertial properties when compared to those of its parent object. In other words, a

crewperson attached to the seat must have larger spring constants than, let's say, a parachute that is also mounted on the seat, since the crewperson has the larger inertial properties of the two objects. The user should ensure that the spring terms are large enough so that very little deflection is required to impart the force required to drive the attached object along with the escape system.

The example given in Section VI can be used as a basis for establishing the appropriate spring and damping constants. The next step is to execute the analysis program through a STEADY STATE task, and then investigate the natural frequencies and damping ratios to ensure they are within tolerances. If they are not, the integrator could have difficulties with the system during a simulation. Therefore, the damping constants and spring terms should be manipulated until reasonable results are obtained. Due to the nature of a complex model, such as the one shown in Section VI, there could be some low damping ratios that are very difficult to eliminate. Note that both components CE and SE contain the human spine model, whose 0.2240 damping ratio cannot be manipulated by the user.

After a simulation is made with a variable step integrator, a time step limitation count is printed for each model state. A time step limitation occurs when the integrator encounters an extreme rate change. If this should happen, the integrator reduces its timestep and performs a recovery process to ensure simulation accuracy. However, this can significantly increase the central processor time required for the simulation. Consequently, many of the EASIEST components have schemes to prevent large changes in rates. For example, the catapult force can be decayed over a time period specified by the user, instead of abruptly being set to zero at stripoff. Specific information on components which have this capability is presented in Section IV.1.

The approach the analyst takes to construct a complex system model can influence the amount of time it requires. Perhaps the most efficient method is to assemble the model a few components at a time, modifying the

model and analysis input files during each design interation to accommodate the components being added. As an example, the user could construct a model using only the sled, rails, and seat components. Once this small model is verified by the various analysis capabilities available in the EASY program, a crewperson component could then be added, and the checkout process repeated. This approach lends itself to correcting problems as they occur, as well as building better designed models.

4. SEQUENCING AN EASIEST MODEL

During the operation of an ejection seat, a variety of discrete events occur that mark transition points in the ejection sequence. Examples include the ignition of a rocket, the burnout of a rocket, and the separation of one object from another. Each such event occurs when either some timing device within the ejection seat triggers it, an event that occurred in some other part of the seat caused a physical switch to be thrown wich triggers it, or the event is defined by the physical status of all or part of the ejection seat and that status has been attained. For example, the seat leaving the guide rails can trigger the sustainer rocket ignition, or the deployment of a parachute can trigger the sustainer rocket ignition, or the deployment of a parachute may be triggered by time in one type of seat design, or by seat speed and/or altitude in another.

In order to allow the EASIEST program to be used in modeling many types of ejection seats, a flexible system has been developed for simulating this event triggering. The fundamental elements of this system are:

a. If an event occurs in one component and knowledge of this occurrence is required by other components, the component in which the event occurs is provided with an output which is:

   (1) Set equal to zero if the event has not yet occurred (or in some cases, has occurred but is no longer occurring)

(2) Set equal to one if the event has occurred (or in some cases is now occurring)

This type of flag is called an "event triggered flag."

b. If an event inside a component is triggered by something outside the component (including time), then that component has been provided with an input which must be:

(1) Set equal to zero if the event is not to begin (or, if occurring, should stop)

SECTION V

PROCEDURES FOR INSTALLING AND SUBMITTING AN EASIEST RUN

The purpose of this section is to explain the EASY5 installation procedure on the ASD computer, and how to submit an EASIEST run.

1. INSTALLING THE EASY5 PROGRAM

The source code for the EASY5 computer program and the EASIEST standard components was delivered to AFWAL/FIER on tape LØ2377. This tape contains 17 files in the following order (the volume and section where the corresponding file listing resides is given in parenthesis where appropriate):

1. <u>EZSTPRC</u> - EASIEST procedure file (Volume 1, Appendix F)

2. <u>BACOMPS</u> - Source for the EASY5 standard components (not EASIEST), associated subroutines and functions (Volume 2, Section III.5)

3. <u>COMPASS</u> - Assembly Language Utility Program (Volume 2, Section II.5)

4. <u>EZSTFTN</u> - Source for the EASIEST standard components, associated subroutines and functions (Volume I, Appendicies G and H)

5. <u>FILOADS</u> - Source for FILOAD. (Volume II, Section II.4)

6. <u>FILDAT</u> - Input data for FILOAD (Volume I, Appendix I)

7. EASYS - Source for the Model Generation program (Volume 2, Section II.1)

8. EASY5 - Relocatables for the Model Generation Program

9. NONSIMS - Source for the Analysis Program (Volume 2, Section II.2)

10. NONSIM5 - Relocatables for the Analysis Program

11. NSMPPTS - Source for the Printer Plot Program (Volume 2, Section III.3)

12. AEROMED - Source for the EASIEST Aeromedical post processor (Volume 1, Appendix E)

13. F4EMAN - EASIEST F-4E aerodynamic maneuvering coefficients (Volume 1, Appendix J)

14. MCORR - Model description for the example in Volume 1, Section VI

15. ACORR - Analysis file for the example in Volume 1, Section VI

16. MODAPP - Model description for the example in Volume 1, Appendix N

17. ANALAPP - Analysis file for the example in Volume 1, Appendix N.

To execute the procedure to install the entire EASY5/EASIEST package from the delivery tape, route the following deck to the ASD computer input queue after instructing the tape library to mount tape number LØ2377:

```
EZ5,T300,I01000,CM100000,NT1.D790183,EASIEST TAPE RUN
REQUEST,TAPE,NT,PE,VSN=LØ2377.
COPYBF,TAPE,TEMP.
BEGIN,EZSTGEN,TEMP,TPW = password for proprietary EASY source files.
```

This procedure will unload the delivery tape, compile the source programs, and catalog all necessary files. In addition, a sample EASIEST run will be submitted, using the same model description and analysis files as the example in Volume I, Section VI.

2. PROCEDURE FOR SUBMITTING AN EASIEST RUN

The following method provides a simple procedure for submitting an EASIEST run into the batch input queue of the ASD computer:

1. Prepare the EASIEST model description file and the EASIEST analysis file as described in the previous section. These files should be stored on your account as permanent files.

2. From an ASD Intercom terminal, which is in the command mode, attach the EASIEST Procedure file using the following command:

   ATTACH,EZSTPRC.

   To perform correctly, this file must be attached with local file name EZSTPRC.

3. To initiate the procedure, type:

   BEGIN, SUBRUN,EZSTPRC,mfname,afname,TIME=t,INOUT=i,CORE=c,
   COEF=j,NOLIST,AEROMED..

   where:

   a. "mfile" is the name of the permanent file containing your model description (this entry is required),

b. "afname" is the name of the permanent file containing your analysis data (this entry is required),

c. "t" is the cpu time in seconds to be allocated for this run (this entry is required only if you wish the allocation to differ from the default of t=100),

d. "i" is the input-output time in seconds to be allocated for this run (this entry can occur anywhere in the BEGIN statement after afname and is required only if you wish the allocation to differ from the default of i=100),

e. "c" is the cpu core space in octal to be allocated for this run (this entry can occur anywhere after afname and is only required is you wish the allocation to differ from the default of c=115000),

f. "j" is the name of the permanent file which contains the aerodynamic coefficients for the EASIEST airplane. If the airplane is not included in the model, "COEF=j" should not be entered.

g. If entered, "NOLIST" deletes the FTN listing from the SUBRUN procedure. Do not include this entry if you wish the FTN listing to be written to output.

h. If specified, "AEROMED" executes the aeromedical post-processor. To suppress execution, do not include this entry.

SECTION VI
EJECTION SEAT ANALYSIS EXAMPLE

This section presents an example of an ejection seat simulation for a model that was assembled using the EASIEST components. All of the EASIEST components were employed in this model, with the exception of AE (Airplane), CS (Airplane Control Surfaces), DR (DART), and AP (Aerodynamic Plate). The implementation of these four components into a model is demonstrated in Appendix N, which also includes a thrust vector control system that was added by using the FORTRAN STATEMENTS command.

Figure 5 presents the model description file used to define the escape system model. The instructions on how this file was assembled are given in Section II. Figure 6 shows the flow chart that was constructed by the Model Generation Program from the instructions contained in the model file. Figure 7 contains the analysis file that was used to define the input tables and parameters. It also initializes the states, and contains the commands that dictate how the model is to be analyzed. An explanation of the commands used by this file is presented in Section III. Figures 8 and 9 show the respective outputs of the steady state analysis and the simulation analysis. Printer plots are shown in Figure 10.

```
MODEL DESC=  SAFEST/EASIEST CORRELATION MODEL
LIST STANDARD COMPONENTS
LOCATION=010   AQ
*
LOCATION=009   FORT
                              ADD VARIABLES=IGFLAG
                              ADD PARAMETERS=CTTIME
FORTRAN STATEMENTS
       IGFLAG=0
       IF(TIME.GE.CTTIME)IGFLAG=1
*
LOCATION=005   CT       INPUTS=SL,SE(SRP=SRP,UST=UST,EST=EST,WST=WST)
                                  FORT(IGFLAG=SW)
LOCATION=027   SR       INPUTS=CT
LOCATION=048   WB       INPUTS=SR
LOCATION=001   SL
LOCATION=041   RL       INPUTS=SL,SE(SRP=SRP,UST=UST,EST=EST,WST=WST)
*
LOCATION=187   FORT
                              ADD VARIABLES=SPFLAG
                              ADD PARAMETERS=SPTIME
FORTRAN STATEMENTS
       SPFLAG=0
       IF(TIME-CTTIME.GE.SPTIME) SPFLAG=1.0
*
LOCATION=184   SP       INPUTS=SE(WST=WST),FORT(SPFLAG=FL)
*
LOCATION=309   FORT
                              ADD VARIABLES=MPFLAG
                              ADD PARAMETERS=MPTIME
FORTRAN STATEMENTS
       MPFLAG=0
       IF(TIME-CTTIME.GE MPTIME) MPFLAG=1.0
*
LOCATION=359   MP       INPUTS=SE(SRP=SRP,UST=UST,EST=EST,WST=WST)
                                  PCRC,FORT(MPFLAG=SW)
LOCATION=355   LIRC     INPUTS=MP(FL=FL),PCRC
                                  CE(XCP=XDO,UCP=UDO,WCP=WDO,ECP=EDO)
LOCATION=318   PCRC
*
LOCATION=209   FORT
                              ADD VARIABLES=GPFLAG
                              ADD PARAMETERS=GPTIME
FORTRAN STATEMENTS
       GPFLAG=0
       IF(TIME-CTTIME GE GPTIME) GPFLAG=1.0
*
LOCATION=258   GP       INPUTS=SE(SRP=SRP,UST=UST,EST=EST,WST=WST)
                                  PCDC,FORT(GPFLAG=SW)
*
LOCATION=203   FORT
                              ADD VARIABLES=DCFLAG
                              ADD PARAMETERS=DCTIME
FORTRAN STATEMENTS
       DCFLAG=0
       IF(TIME-CTTIME GE MPTIME+DCTIK ) DCFLAG=1
*
LOCATION=253   LIDC     INPUTS=GP(FL=FL),PCDC
                                  SE(SRP=XDO,UST=UDO,WST=WDO,EST=EDO)
                                  FORT(DCFLAG=OFF)
LOCATION=216   PCDC
LOCATION=023   AS       INPUTS=RL,SE(SRP=SRP,UST=UST,EST=EST,WST=WST)
                                  SR(RON=RON)
```

Figure 5. Model Generation Program Input File

```
*
LOCATION=119   FORT
                                ADD VARIABLES=CEFLAG
                                ADD PARAMETERS=CETIME
FORTRAN STATEMENTS
       CEFLAG=0
       IF(FTSRL.EQ.1.  .AND. TIME-TTSRL.GE.CETIME) CEFLAG=1.0
*
LOCATION=117   RSCS     INPUTS=SE(SRP=XPB,UST=UPB,EST=EPB,WST=WPB)
                               CE(XCP=XAB,UCP=UAB,ECP=EAB,WCP=WAB)
                               FORT(CEFLAG=FL)
LOCATION=143   RSKC     INPUTS=CE(XCP=XPB,UCP=UPB,ECP=EPB,WCP=WPB)
                               ABSK
LOCATION=147   CE       INPUTS=LIRC,RSCS,RSKC(FPB=FAU,TPB=TAU)
                               FORT(CEFLAG=SW)
LOCATION=113   ABSK
LOCATION=045   SE       INPUTS=CT(1=1),SR(1=2),SP(1=3),OP(1=4),MP(1=5)
                               RL(1=1),AS(1=2),LIDC(FDO=F2.3,TDO=T2.3)
                               RSCS(FPB=F2.4,TPB=T2.4),WB
*
LOCATION=067   AM       INPUTS=CE
END OF MODEL
PRINT
```

Figure 5. (Continued)

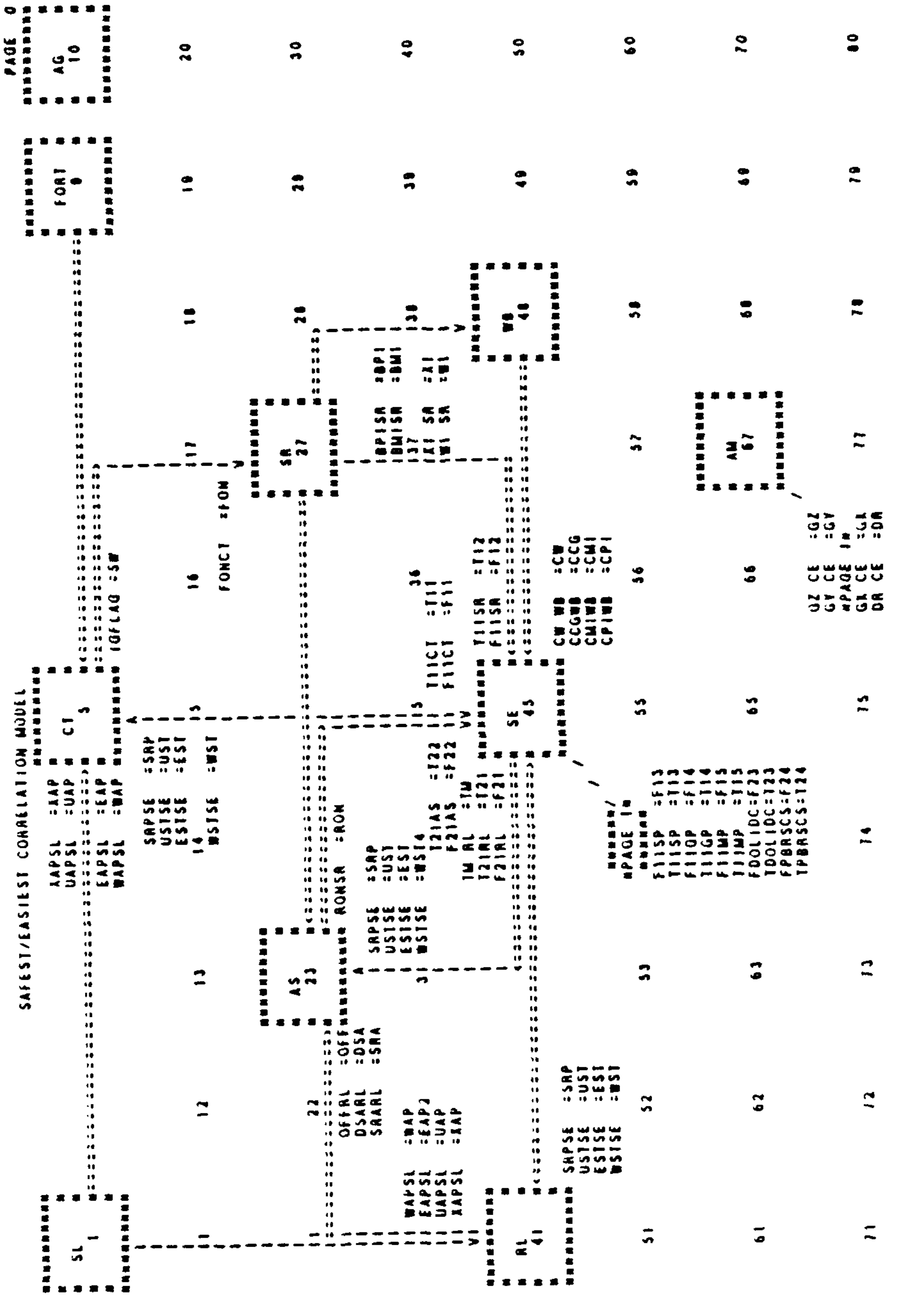


Figure 6. EASY Schematic

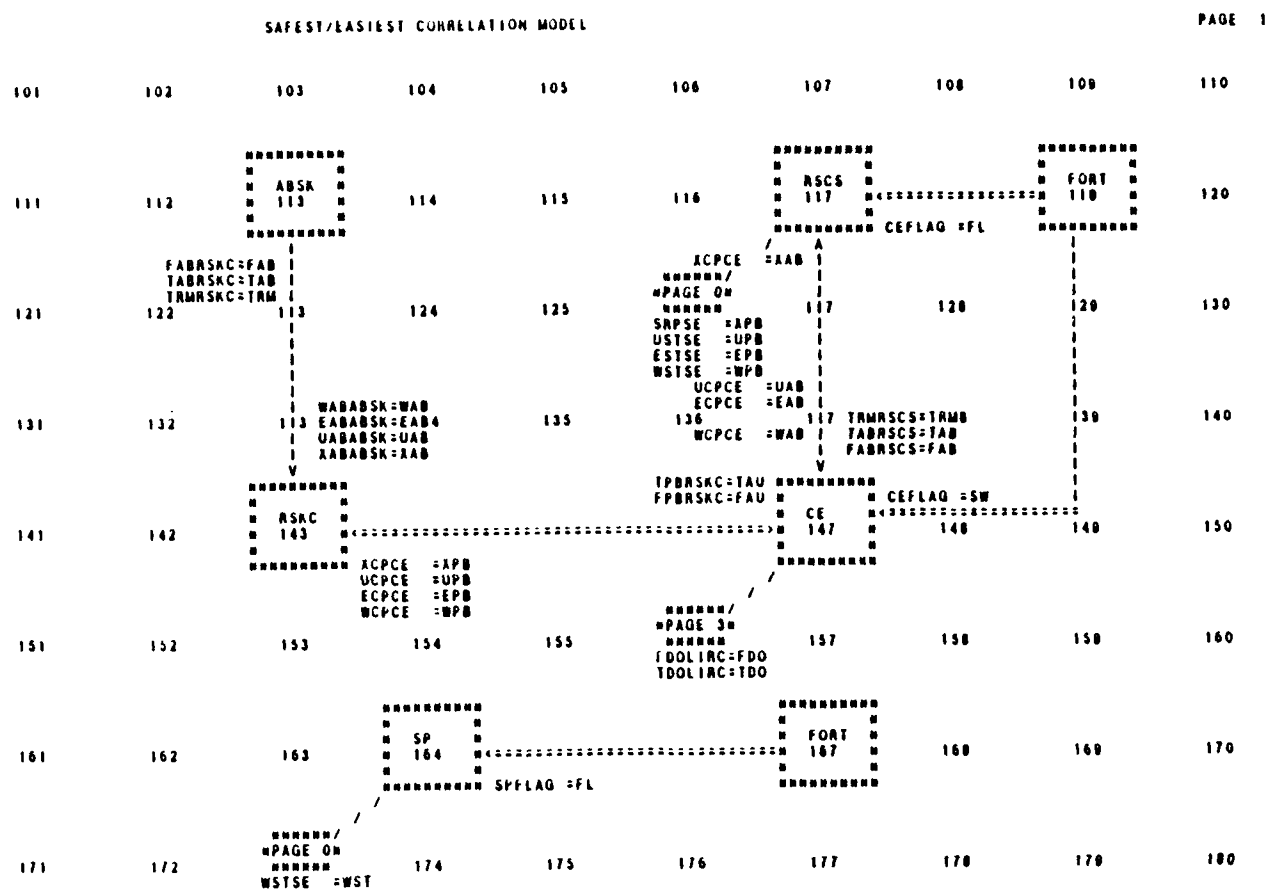


Figure 6. (Continued)

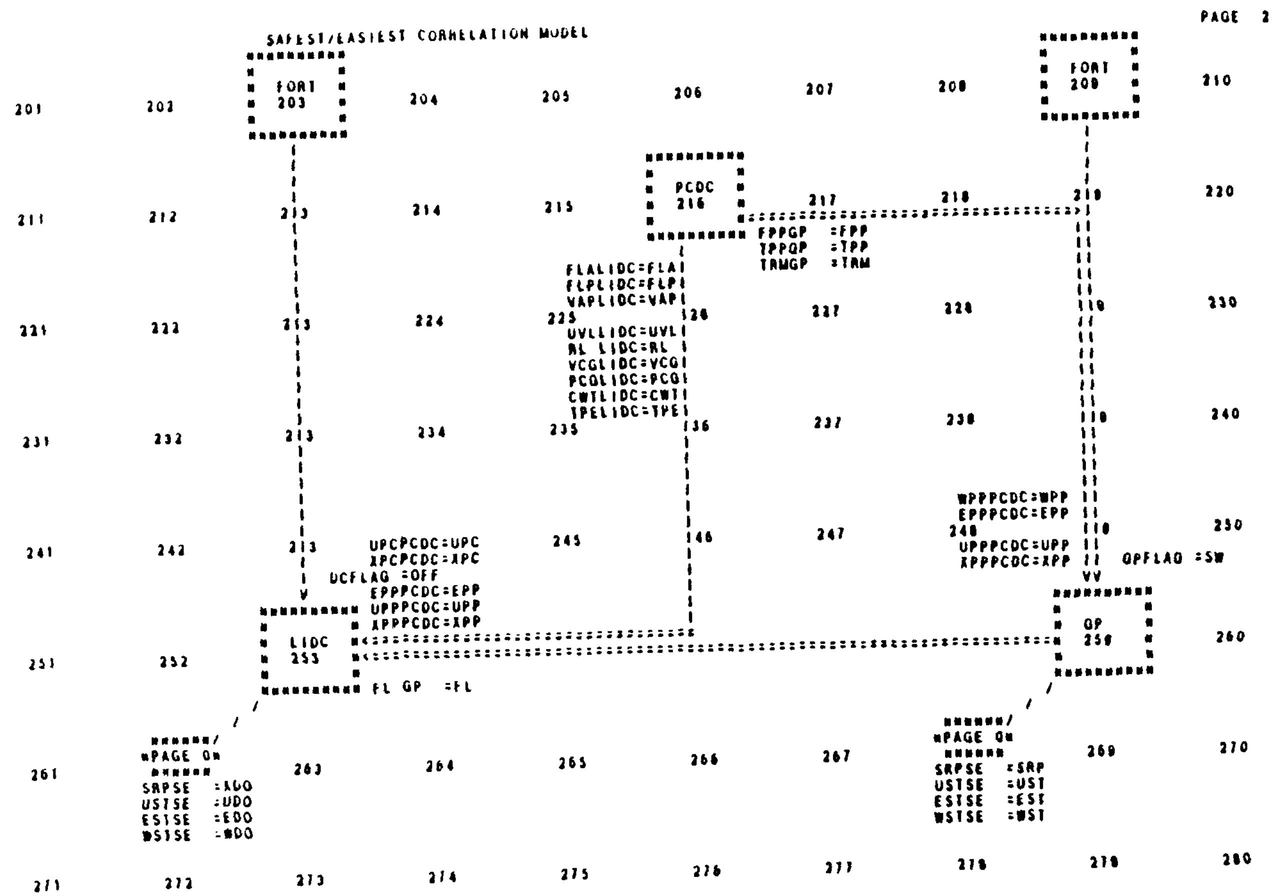


Figure 6. (Continued)

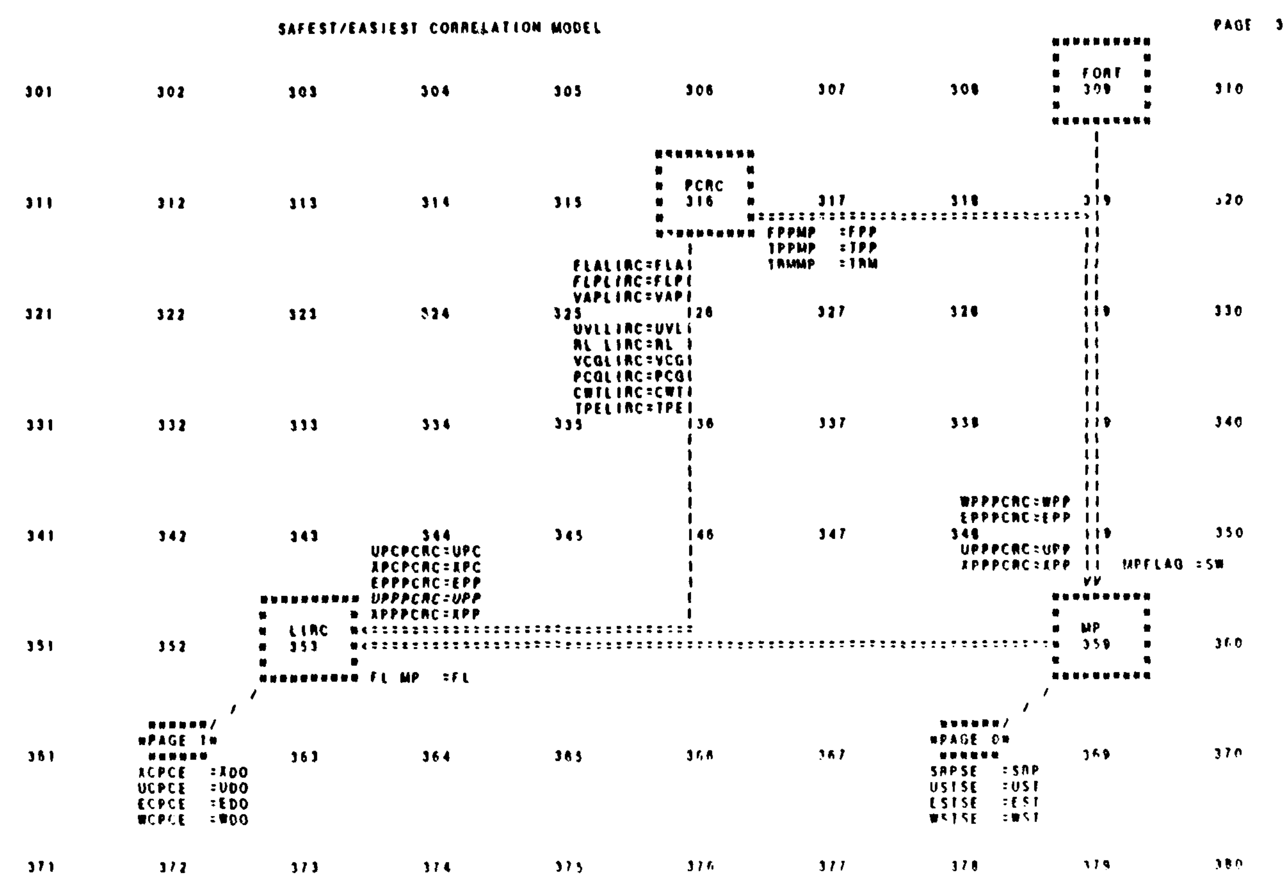


Figure 6. (Continued)

```
*  ------  EASIEST EJECTION SEAT ANALYSIS FILE  ------
*
*
*  ***********************************************
*  ******************  TABLES  *******************
*  ***********************************************
*
*
*  -----  TABLE FOR COMPONENT "CT" (CATAPULT)  -----
*
TABLE      TCPCT,10
*  CATAPULT PROPELLANT WEB CONSUMED (INCHES)
0.0         8.319E-3   1.502E-2   3.086E-2   4.986E-2
7.638E-2   1.081E-1   1.172E-1   1.264E-1   1.422E-1
*  CATAPULT PROPELLANT CONSUMED (SLUGS)
0.0         1.116E-4   1.960E-4   4.018E-4   6.249E-4
8.770E-4   1.231E-3   1.347E-3   1.472E-3   1.703E-3
*
*  -----  TABLE FOR COMPONENT "SR" (SUSTAINER ROCKET)  -----
*
TABLE      TRFSR,7
*  TIME INTO SUSTAINER ROCKET BURN (SEC)
0.0        .001       .030       .140       .260       .310       .350
*  SUSTAINER ROCKET FORCE (LBS)
0.0        3425.      3630.      3400.      3150.      1000.      0.00
*
*  -----  TABLE FOR COMPONENT "AS" (SEAT AERO)  -----
*
TABLE      TAEAS,9
*  CREWMAN EXPOSED LENGTH (FT)
0.00    0.25    0.50    0.75    1.00    4.25    4.75    5.25    5.50
*  CREWMAN EXPOSED AREA (FT**2)
0.00    0.15    0.40    0.63    0.94    6.26    6.55    6.85    6.96
*
*  -----  TABLES FOR COMPONENT "SP" (STAPAC)  -----
*
TABLE      TRTSP,7
*  TIME INTO VERNIER ROCKET BURN (SEC)
0.0        0.005      0.015      0.045      0.160      0.330      0.350
*  VERNIER ROCKET FORCE (LBS)
0.0        856.0      825.0      725.0      700.0      650.0      0.0
*
TABLE      TMASP,3
*  VALUES OF GIMBAL ANGLE (DEG)
-35.0      0.0        35.0
*  VALUES OF MECHANICAL ADVANTAGE (--)
-1.5       -1.5       -1.5
*
TABLE      TSTSP,3
*  VALUES OF GIMBAL ANGLE (DEG)
-35.0      0.0        35.0
*  VALUES OF SPRING TORQUE (FT-LBS)
0.0        0.0        0.0
*
*  -----  TABLE FOR COMPONENT "GP" (DROGUE CHUTE GUN)  -----
*
TABLE,TMFGP,4
*  TIME SINCE GUN INITIATION (SEC)
0.0        0.0001     0.002      .0021
*  DROGUE GUN FORCE (LBS)
0.0        3528.      3528.      0.0
*
*  -----  TABLE FOR COMPONENT "PCDC" (DROGUE CHUTE)  -----
*
TABLE,TCWLIDC,8
```

Figure 7. Analysis Program Input File

```
*  LENGTH OF STRETCHED DROGUE CHUTE (FT)
0.0       8.00      15.0      22.0      23.0      24.0      25.0      26.0
*  DROGUE CHUTE WEIGHT EXTRACTED FROM THE PACK (LBS)
0.0       1.20      2.40      3.52      4.58      5.64      6.69      7.75
*
*  -----  TABLE FOR COMPONENT "MP" (RECOVERY CHUTE MORTAR)  -----
*
TABLE     TMPMP,10
*  MORTAR PROPELLANT WEB CONSUMED (INCHES)
0.0         8.319E-3    1.502E-2    3.086E-2    4.986E-2
7.636E-2    1.081E-1    1.172E-1    1.264E-1    1.422E-1
*  MORTAR PROPELLANT CONSUMED (SLUGS)
0.0         5.600E-5    9.800E-5    2.010E-4    3.125E-4
4.385E-4    6.155E-4    6.735E-4    7.360E-4    8.515E-4
*
*  -----  TABLE FOR COMPONENT "PCRC" (RECOVERY CHUTE)  -----
*
TABLE     TCWLIRC,8
*  LENGTH OF STRETCHED RECOVERY CHUTE (FT)
0.0    3.00   24.00   28.00   30.00   32.00   34.00   38.00
*  RECOVERY CHUTE WEIGHT EXTRACTED FROM THE PACK (LBS)
0.0    3.00    8.40   12.06   13.63   15.01   16.20   18.00
*
****************************************************************
********************  PARAMETER VALUES  ************************
****************************************************************
*
*
*  -----  PARAMETERS FOR COMPONENT "CT" (CATAPULT)  -----
*
PARAMETER VALUES
CTTIME=.001
SAPCT=-.65,0,-3.2                     AAPCT=-.5716,0,1.593
UCLCT=4.2208     CSKCT=2.833         VI CT=19.5          PA CT=0.785
PT CT=10         CBPCT=9000          C  CT=.001703       CI CT=.00001
PMWCT=25.5       SK CT=100000        CK CT=150            GAMCT=1.199
TF CT=2800       C1 CT=.120          C2 CT=0.3           B  CT=.0335
BXPCT=.44        TI CT=294           UP CT=1             TDECT=0
*
*  -----  PARAMETERS FOR COMPONENT "SR" (SUSTAINER ROCKET)  -----
*
PCGSR=-.51,0,-1.58                    EA SR=0,4.53,0
XRNSR=.0151,0,1.5952
YAWSR=0          PITSR=-53.78        PL SR=2.917         PODSR=0.2107
PIDSR=0.1458
*
*  ----- PARAMETERS FOR COMPONENT "WB" (WEIGHT AND BALANCE)  -----
*
AB WB=1          SW WB=91.5          SX WB=-.0276,.0431,-.6005
SM WB=4.1497,4.4476,2.4538           SP WB=0.1311,1.1391,-0.3812
*
*  -----  PARAMETERS FOR COMPONENT "RL" (RAILS)  -----
*
BL1RL =  -0.4125    0.6670    0.6333
BL2RL =  -0.4713    0.6670   -0.1146
BL3RL =  -0.5611    0.6670   -1.2561
BL4RL =  -0.4125   -0.6670    0.6333
BL5RL =  -0.4713   -0.6670   -0.1146
BL6RL =  -0.5611   -0.6670   -1.2561
RLRRL=-4.041     XRRRL=-0.5716, 0.667,1.593
RLLRL=-4.041     XRLRL=-0.5716,-0.667,1.593
ERLRL=0,17,0
SPRRL=50000,50000          DPGRL=40,40
SBFRL=0.02                 ZTSRL=-3.0
*
```

Figure 7. (Continued)

```
#  -----  PARAMETERS FOR COMPONENT "CE" (CREWPERSON) -----
#
CETIME=1.762
PC CE=95           CEWCE=49.8
CMICE=10.3883,10.5159,3.0416      CPICE=0.0874,2.0290,0.0311
CLPCE=-4.54E-3    CMQCE=-1.05E-2   CNRCE=-5.24E-4
XSPCE=-.7204,-.0197,1.0636
#
#  -----  PARAMETERS FOR COMPONENT "RSCS" (MAN TO SEAT RESTRAINTS)  -----
#
XYZRSCS=.7204,.0197,-1.0636       EA RSCS=Z
XR RSCS=27900                     XD RSCS=300
ER RSCS=1750,1750,1750            ED RSCS=15,15,9
#
#  -----  PARAMETERS FOR COMPONENT "ABSK" (SURVIVAL KIT)  -----
#
WT ABSK=25.4
BMIABSK=0.4409,0.5774,0.4401        BPIABSK=0.1841,0.1860,0.2079
#
#  -----  PARAMETERS FOR COMPONENT "RSKC" (KIT/CREWP RESTRAINTS)  -----
#
FL RSKC=0          XYZRSKC=-.1216,.0016,1.3841        EA RSKC=Z
XR RSKC=6200    XD RSKC=120
ER RSKC=750,750,750          ED RSKC=1.8,2.2,1.8
#
#  -----  PARAMETERS FOR COMPONENT "AS" (SEAT AERODYNAMICS)  -----
#
ZWSAS=-3.50        XEMAS=0,0,-3.25                    CDXAS=0
ECXAS=1            ECYAS=1            ECZAS=1         CLPAS=-4.538E-4
CMQAS=-1.047E-2                       CNRAS=-5.236E-4
S  AS=6.94
#
#  -----  PARAMETERS FOR COMPONENT "SP" (STAPAC)  -----
#
SPTIME=0.166
YPRSP=2            AVWSP=28650        WMISP=.00147     SMISP=.001
RIISP=.000636      RIFSP=.00051       XR SP=.2084,0,0.7
GSASP=-30          GSFSP=30           SPRSP=5.0        DPGSP=.015
FMTSP=800          TMXSP=.52          TNFSP=.104       TOSSP=0
#
#  -----  PARAMETERS FOR COMPONENT "PCRC" (RECOVERY CHUTE)  -----
#
STIPCRC=461.8      RSCPCRC=72.64      RFMPCRC=1        RFDPCRC=1.15
RFSPCRC=46         B  PCRC=.011       CI PCRC=3.33
CT PCRC=10.41,89.10,-171.5            CN PCRC=-38.2,37.0,39.7
CM PCRC=0.0        FD PCRC=1          PWTPCRC=20.19
PMIPCRC=0.3052,0.2577,0.3490          PPIPCRC=0.0546,0.1464,0.0975
TEMPCRC=.05        CSPPCRC=2000       CDPPCRC=14
DPGPCRC=2,2,2
#
#  -----  PARAMETERS FOR COMPONENT "LIRC" (REC CHUTE LINES)  -----
#
OFFLIRC=0
FTRLIRC=1          BLILIRC=1          APXLIRC=Z        APILIRC=0,0,-2.59
FSOLIRC=10         ULLLIRC=550        ULSLIRC=.35
GORLIRC=28         TYPLIRC=2
#
#  -----  PARAMETERS FOR COMPONENT "MP" (PARACHUTE MORTAR)  -----
#
MPTIME=1.762
XYZMP=-0.2777,0.0177,-2.4725          EA MP=0,0,0
XR MP=6200         XD MP=90
ER MP=750,750,750                     ED MP=2,1.6,2
UV MP=-.0604,.0399,-.9974             CSKMP=1.4583
VI MP=5.75                            PA MP=.5185
```

Figure 7. (Continued)

```
PT MP=10            CBPMP=9000         C  MP=8.515E-4
CI MP=.00001        PMWMP=25.5         GAMMP=1.199        TF MP=2600
C1 MP=.120          C2 MP=.300         B  MP=.0335        BXPMP=.44
TI MP=294           TDEMP=.05
*
*  -----  PARAMETERS FOR COMPONENT "PCDC" (DRAG CHUTE)  -----
*
STIPCDC=10.57     RSCPCDC=19.63      RFMPCDC=0          RFDPCDC=0
RFSPCDC=0         B  PCDC=.08        CI PCDC=1
CT PCDC=-0.546,-24.4,14.69           CN PCDC=31.2,-3.11,-1.97
CM PCDC=-0.64,0.2                    FD PCDC=1          PWTPCDC=8.1
PMIPCDC=0.0340,0.0380,0.0140         PPIPCDC=Z
TEMPCDC=.05       CSPPCDC=2000       CDPPCDC=14
DPGPCDC=2,2,2
*
*  -----  PARAMETERS FOR COMPONENT "LIDC" (DRAG CHUTE LINES)  -----
*
DCTIME=0.15
BLILIDC=2          APXLIDC=-3.84,0,-1.04
APILIDC=-.55,.51,-1.04               AP2LIDC=-.55,-.51,-1.04
FSOLIDC=10         ULLLIDC=2000      ULSLIDC=.4
GORLIDC=16          FTRLIDC=0.50      TYPLIDC=1
*
*  -----  PARAMETERS FOR COMPONENT "GP" (DRAG CHUTE MORTAR)  -----
*
GPTIME=.180
UV GP=-.6189,-.342,-.7071            XMOGP=-.45,-.208,-1.345
XYZGP=-.32,-.33,-.70                 EA GP=Z
XR GP=2500       XD GP=30
ER GP=500,500,500                    ED GP=.8,.8,.45
TDEGP=0
*
*  -----  PARAMETERS FOR COMPONENT "AG" (ATMOSPHERE AND GRAVITY)  -----
*
SW AG=1          WINAG=Z
BP AG=25.76      TE AG=73.8      H  AG=0
*
*  -----  PARAMETERS FOR COMPONENT "AM" (AEROMEDICAL)  -----
*
FL AM=1.
*
**** INITIALIZATION
*
INITIAL CONDITIONS=UAPSL=821,0,0,XAPSL=0,0,0
                   W1 SR=5.6
                    SRPSE=-.416,0,.526,USTSE=801.65,0,177.2
                    ESTSE=0,12.5,0,WSTSE=Z
                    ESGSP=0,-12.38,0   ESRSP=0,-14.0,0
CALC XIC
INT CONTROLS    XAPSL=Z,UAPSL=Z,EAPSL=Z,WAPSL=Z
                EF CT=0,EL CT=0,WK CT=0,WB CT=0
                EF MP=0,EL MP=0,WK MP=0,WB MP=0
                W1 SR=0,EC LIDC=0,TF LIDC=0
                EC LIRC=0,TF LIRC=0
                ESGSP=Z,ESRSP=Z,WG SP=0
SS ITERATIONS=60
STEADY STATE
XIC-X
PARAMETER VALUES
DPGRL=0,0
ALL STATES
*
***  SIMULATION
*
DEFINE RATES   R43=UDST
```

Figure 7. (Continued)

```
                    R45=WDST
PRINT CONTROL=5
PRINT VARIABLES
UDST(1),UDST(2),UDST(3)
USTSE(1),USTSE(2),USTSE(3)
SRPSE(1),SRPSE(2),SRPSE(3)
WDST(1),WDST(2),WDST(3)
WSTSE(1),WSTSE(2),WSTSE(3)
ESTSE(1),ESTSE(2),ESTSE(3)
RADAM,DREAM,CF CT
UCPCE(1),UCPCE(2),UCPCE(3)
XCPCE(1),XCPCE(2),XCPCE(3)
ECPCE(1),ECPCE(2),ECPCE(3)
XPCPCRC(1),XPCPCRC(2),XPCPCRC(3)
XPCPCDC(1),XPCPCDC(2),XPCPCDC(3)
PRINTER PLOTS
TITLE=EASIEST EJECTION SEAT MODEL
DISPLAY1,UDST(1),VS,TIME
         UDST(2),VS,TIME
         UDST(3),VS,TIME
         USTSE(1),VS,TIME
         USTSE(2),VS,TIME
DISPLAY2,USTSE(3),VS,TIME
         SRPSE(1),VS,TIME
         SRPSE(2),VS,TIME
         SRPSE(3),VS,TIME
         WDST(1),VS,TIME
DISPLAY3,WDST(2),VS,TIME
         WDST(3),VS,TIME
         WSTSE(1),VS,TIME
         WSTSE(2),VS,TIME
         WSTSE(3),VS,TIME
DISPLAY4,ESTSE(1),VS,TIME
         ESTSE(2),VS,TIME
         ESTSE(3),VS,TIME
         RADAM,VS,TIME
         DREAM,VS,TIME
DISPLAY5,XCPCE(1),VS,TIME
         XCPCE(2),VS,TIME
         XCPCE(3),VS,TIME
         ECPCE(1),VS,TIME
         ECPCE(2),VS,TIME
DISPLAY6,ECPCE(3),VS,TIME
         XPCPCDC(1),VS,TIME
         XPCPCDC(2),VS,TIME
         XPCPCDC(3),VS,TIME
INT MODE=2,TMAX-5.00,TINC=.001
PRATE=1,OUTRATE=10
PRATE2=10,OUTRATE2=10,PRINT2 = 5,PRINT2 FROM,0.30, 10, 5.0
SIMULATE
```

Figure 7. (Continued)

```
/*/*/*/   STEADY STATE ANALYSIS   /*/*/*/
A MAXIMUM OF   60 ITERATIONS CAN BE USED

TIME = 0.          --------------------------------------------------------   CASE NO.   1

                                        STATES

  1 EF CT    = 0.           2 EL CT    = 0.           3 WK CT    = 0.           4 WB CT    = 0.           5 WI SR    =  5.6000
UAPSL  ( 1) =  821.00     UAPSL  ( 2) = 0.           UAPSL  ( 3) = 0.
XAPSL  ( 1) = 0.          XAPSL  ( 2) = 0.           XAPSL  ( 3) = 0.
WAPSL  ( 1) = 0.          WAPSL  ( 2) = 0.           WAPSL  ( 3) = 0.
EAPSL  ( 1) = 0.          EAPSL  ( 2) = 0.           EAPSL  ( 3) = 0.
 18 WG SP   = 0.
ESGSP  ( 1) = 0.          ESGSP  ( 2) = -12.380      ESGSP  ( 3) = 0.
ESRSP  ( 1) = 0.          ESRSP  ( 2) = -14.000      ESRSP  ( 3) = 0.
 25 EF MP   = 0.           26 EL MP   = 0.            27 WK MP   = 0.           28 WB MP   = 0.           29 EC LIRC = 0.
 30 TF LIRC = 0.
UPPPCRC( 1) =  821.00     UPPPCRC( 2) =  .22426E-16 UPPPCRC( 3) =  .17685E-18
XPPPCRC( 1) = -1.2181     XPPPCRC( 2) =  .17788E-01 XPPPCRC( 3) = -1.8254
WPPPCRC( 1) = 0.          WPPPCRC( 2) = 0.          WPPPCRC( 3) = 0.
EPPPCRC( 1) =  .73738E-03 EPPPCRC( 2) =  12.409     EPPPCRC( 3) =  .23598E-02
UPCPCRC( 1) =  821.00     UPCPCRC( 2) = -.10115E-16 UPCPCRC( 3) =  .26379E-18
XPCPCRC( 1) = -1.2181     XPCPCRC( 2) =  .17788E-01 XPCPCRC( 3) = -1.8249
 49 EC LIDC = 0.           50 TF LIDC = 0.
UPPPCDC( 1) =  821.00     UPPPCDC( 2) = -.45449E-20 UPPPCDC( 3) =  .22589E-18
XPPPCDC( 1) = -.87956     XPPPCDC( 2) = -.32998     XPPPCDC( 3) = -.85219E-01
WPPPCDC( 1) = -.48101E-18 WPPPCDC( 2) = -.61689E-18 WPPPCDC( 3) = -.22183E-18
EPPPCDC( 1) =  .73738E-03 EPPPCDC( 2) =  12.409     EPPPCDC( 3) =  .23599E-02
UPCPCDC( 1) =  821.00     UPCPCDC( 2) =  .67369E-20 UPCPCDC( 3) =  .16738E-19
XPCPCDC( 1) = -.87956     XPCPCDC( 2) = -.32998     XPCPCDC( 3) = -.84719E-01
UCPCE  ( 1) =  801.83     UCPCE  ( 2) = -.33010E-02 UCPCE  ( 3) =  176.39
XCPCE  ( 1) =  .58399E-01 XCPCE  ( 2) =  .19747E-01 XCPCE  ( 3) = -.65722
WCPCE  ( 1) =  .13280E-18 WCPCE  ( 2) = -.29004E-18 WCPCE  ( 3) =  .13392E-18
ECPCE  ( 1) =  .74230E-03 ECPCE  ( 2) =  12.406     ECPCE  ( 3) =  .23828E-02
 81 SCDCE   = -.31913E-19  82 SC CE   =  .10781E-18
UABABSK( 1) =  801.83     UABABSK( 2) = -.33010E-02 UABABSK( 3) =  176.39
XABABSK( 1) =  .23700     XABABSK( 2) =  .21291E-01 XABABSK( 3) =  .72478
WABABSK( 1) = -.17218E-18 WABABSK( 2) = -.98839E-19 WABABSK( 3) = -.14600E-18
EABABSK( 1) =  .74230E-03 EABABSK( 2) =  12.406     EABABSK( 3) =  .23828E-02
USTSE  ( 1) =  801.82     USTSE  ( 2) = -.32995E-02 USTSE  ( 3) =  176.42
SRPSE  ( 1) = -.41662     SRPSE  ( 2) = -.33568E-05 SRPSE  ( 3) =  .52644
WSTSE  ( 1) = -.31391E-31 WSTSE  ( 2) =  .80908E-30 WSTSE  ( 3) =  .70327E-31
ESTSE  ( 1) =  .73738E-03 ESTSE  ( 2) =  12.409     ESTSE  ( 3) =  .23599E-02
107 SCDSE   =  .11285E-20 108 SC SE   = -.78813E-20

                                        RATES

  1 R1      = 0.           2 R2      = 0.            3 R3      = 0.           4 R4      = 0.           5 R5      = 0.
R6     ( 1) = 0           R6     ( 2) = 0.          R6     ( 3) = 0.
R7     ( 1) = 0.          R7     ( 2) = 0.          R7     ( 3) = 0.
R8     ( 1) = 0.          R8     ( 2) = 0.          R8     ( 3) = 0.
R9     ( 1) = 0.          R9     ( 2) = 0.          R9     ( 3) = 0.
 18 R10     = 0.
R11    ( 1) = 0.          R11    ( 2) = 0.          R11    ( 3) = 0.
R12    ( 1) = 0.          R12    ( 2) = 0.          R12    ( 3) = 0.
 25 R13     = 0.           26 R14     = 0.           27 R15     = 0.           28 R16     = 0.           29 R17     = 0.
 30 R18     = 0.
```

Figure 8. Steady State Output

```
R19      ( 1) =  .13723E-07 R19      ( 2) =  .35411E-07 R19      ( 3) = -.30674E-07
R20      ( 1) = -.90949E-10 R20      ( 2) = -.28309E-09 R20      ( 3) =  .19206E-09
R21      ( 1) =  .65904E-05 R21      ( 2) =  .85426E-06 R21      ( 3) =  .40130E-05
R22      ( 1) = 0.          R22      ( 2) = 0.          R22      ( 3) = 0.
R23      ( 1) = 0.          R23      ( 2) =  .14658E-13 R23      ( 3) = -.15363E-09
R24      ( 1) = -.90949E-10 R24      ( 2) = -.28309E-09 R24      ( 3) =  .19206E-09
 49 R25       = 0.           50 R26       = 0.
R27      ( 1) =  .12027E-07 R27      ( 2) =  .32576E-07 R27      ( 3) = -.23887E-07
R28      ( 1) = -.90949E-10 R28      ( 2) = -.28309E-09 R28      ( 3) =  .19206E-09
R29      ( 1) =  .60659E-05 R29      ( 2) = -.20976E-05 R29      ( 3) =  .24991E-05
R30      ( 1) = -.22716E-18 R30      ( 2) = -.61689E-18 R30      ( 3) = -.96915E-19
R31      ( 1) = 0.          R31      ( 2) = -.50817E-17 R31      ( 3) = -.10745E-10
R32      ( 1) = -.90949E-10 R32      ( 2) = -.28309E-09 R32      ( 3) =  .19206E-09
R33      ( 1) =  .27726E-06 R33      ( 2) = -.90820E-07 R33      ( 3) =  .52068E-07
R34      ( 1) = -.84765E-08 R34      ( 2) =  .54126E-08 R34      ( 3) = -.54043E-08
R35      ( 1) = -.42234E-06 R35      ( 2) = -.26925E-06 R35      ( 3) = -.30832E-06
R36      ( 1) =  .13711E-18 R36      ( 2) = -.29005E-18 R36      ( 3) =  .18225E-18
 81 R37       = -.48828E-07  82 R38       = -.31913E-18
R39      ( 1) =  .30707E-07 R39      ( 2) = -.69990E-07 R39      ( 3) =  .10421E-06
R40      ( 1) =  .10914E-10 R40      ( 2) =  .74610E-12 R40      ( 3) =  .30324E-16
R41      ( 1) =  .59328E-08 R41      ( 2) =  .78995E-08 R41      ( 3) =  .62432E-08
R42      ( 1) = -.14949E-18 R42      ( 2) = -.99833E-19 R42      ( 3) = -.20430E-16
R43      ( 1) = -.55218E-06 R43      ( 2) =  .24509E-06 R43      ( 3) = -.30748E-06
R44      ( 1) =  .90949E-10 R44      ( 2) =  .28309E-09 R44      ( 3) = -.19206E-09
R45      ( 1) =  .10412E-04 R45      ( 2) =  .21368E-04 R45      ( 3) =  .11333E-04
R46      ( 1) =  .72043E-31 R46      ( 2) =  .80908E-30 R46      ( 3) = -.15910E-31
107 R47       =  .30748E-06 108 R48       =  .11285E-29

                                    VARIABLES

  1 VS AG     = 0.            2 RHOAG     = 0.            3 IGFLAG    = 0.            4 FL CT     = 0.            5 FONCT     = 0.
FCICT    ( 1) =  106.31     FCICT    ( 2) = -.10146E-01 FCICT    ( 3) =  368.98
TCICT    ( 1) =  .18183E-01 TCICT    ( 2) =  380.27     TCICT    ( 3) =  .57997E-02
F11CT    ( 1) = -24.536     F11CT    ( 2) = -.42692E-02 F11CT    ( 3) = -381.21
T11CT    ( 1) = -.13661E-01 T11CT    ( 2) = -170.57     T11CT    ( 3) =  .27750E-02
 18 CF CT     = 0.           19 CEXCT     = -.38400E-02  20 CV CT     =  .15938E-08 21 TLOCT     =  24.841      22 PC CT     =  3.1408
 23 R  CT     =  3508.9      24 CVHCT     =  17632.      25 TSOCT     = 0.          26 FSOCT     = 0.           27 PHASR     = 0.
 28 RONSR     = 0.
F11SR    ( 1) = 0.          F11SR    ( 2) = 0.          F11SR    ( 3) = 0.
T11SR    ( 1) = 0.          T11SR    ( 2) = 0.          T11SR    ( 3) = 0.
X1 SR    ( 1) = -.51000     X1 SR    ( 2) = 0.          X1 SR    ( 3) = -1.5600
BM1SR    ( 1) =  .12339     BM1SR    ( 2) =  .12416     BM1SR    ( 3) =  .22494E-02
BP1SR    ( 1) = 0.          BP1SR    ( 2) =  .96587E-02 BP1SR    ( 3) = 0.
 44 FR SR     = 0.           45 PWISR     =  5.6000      46 SPISR     =  .23746E+06 47 RHOSR     =  95.106      48 VWISR     =  4.6318
TMISR    ( 1) =  .22643     TMISR    ( 2) =  .22643     TMISR    ( 3) =  .18667E-02
 52 TIGSR     = 0.           53 CW WB     =  97.100
CCGWB    ( 1) = -.55421E-01 CCGWB    ( 2) =  .40814E-01 CCGWB    ( 3) = -.65584
CMIWB    ( 1) =  4.4244     CMIWB    ( 2) =  4.7609     CMIWB    ( 3) =  2.4946
CPIWB    ( 1) =  .13451     CPIWB    ( 2) =  1.2247     CPIWB    ( 3) = -.37442
F21RL    ( 1) =  110.81     F21RL    ( 2) = -.11881E-01 F21RL    ( 3) = -8.8983
T21RL    ( 1) = -8.9433     T21RL    ( 2) =  287.37     T21RL    ( 3) = -1.5375
FR1RL    ( 1) = -106.31     FR1RL    ( 2) =  .10146E-01 FR1RL    ( 3) =  32.502
TR1RL    ( 1) =  7.1071     TR1RL    ( 2) = -309.79     TR1RL    ( 3) = -.57993E-02
 75 FL RL     = 0.           76 FTSRL     = 0.           77 TTSRL     = 0.          78 OFFRL     = 0.

 79                     DSARL  ( 3, 3)
          1             2             3
   1   .97684      -.40189E-05   .21488
   2   .12569E-04   1.0000       -.38423E-04
   3  -.21488       .40226E-04   .97664
SRARL    ( 1) = -.41682     SRARL    ( 2) = -.33568E-05 SRARL    ( 3) =  .52844
 91 DISRL     =  .67135
TM RL    ( 1) =  821.00     TM RL    ( 2) = 0.          TM RL    ( 3) = 0.
```

Figure 8. (Continued)

```
 95 SPFLAG  = 0.          96 PHASP    = 0.
F11SP  ( 1) = 0.         F11SP  ( 2) = 0.         F11SP  ( 3) = 0.
T11SP  ( 1) = 0.         T11SP  ( 2) = 0.         T11SP  ( 3) = 0.
103 TINSP   = 0.         104 ECASP   = 0.         105 MPFLAG  = 0.         106 FL MP   = 0.
F11MP  ( 1) = -4.3386    F11MP  ( 2) =  .81214E-03 F11MP  ( 3) =  19.718
T11MP  ( 1) =  .35102    T11MP  ( 2) =  16.203    T11MP  ( 3) =  .76567E-01
FPPMP  ( 1) =  .86116E-08 FPPMP  ( 2) =  .22222E-07 FPPMP  ( 3) = -20.190
TPPMP  ( 1) =  .24037E-07 TPPMP  ( 2) = -.82670E-08 TPPMP  ( 3) =  .81508E-08
119 FM MP   = 0.         120 EXMMP   = 0.         121 VM MP   = 0.         122 TLOMP   =  11.090     123 PC MP   =  2.5526
124 R  MP   =  3500.0    125 CVHMP   =  17632.    126 TSOMP   = 0.         127 FSOMP   = 0.
TRMMP  ( 1) =  821.00    TRMMP  ( 2) =  .28309E-09 TRMMP  ( 3) = -.13206E-09
131 FLALIRC = 0.         132 SWILIRC = 0.
FDOLIRC( 1) = 0.         FDOLIRC( 2) = 0.         FDOLIRC( 3) = 0.
TDOLIRC( 1) = 0.         TDOLIRC( 2) = 0.         TDOLIRC( 3) = 0.
FLPLIRC( 1) = 0.         FLPLIRC( 2) = 0.         FLPLIRC( 3) = 0.
FAPLIRC( 1) = 0.         FAPLIRC( 2) = 0.         FAPLIRC( 3) = 0.
VAPLIRC( 1) = 0.         VAPLIRC( 2) = 0.         VAPLIRC( 3) = 0.
148 FLLLIRC = 0.         149 ELMLIRC = 0.         150 ELCLIRC = 0.         151 DEMLIRC = 0.          152 RMNLIRC = 0.
153 DISLIRC = 0
CONLIRC( 1) = 0.         CONLIRC( 2) = 0.         CONLIRC( 3) = 0.         CONLIRC( 4) = 0.
TCGLIRC( 1) = 0.         TCGLIRC( 2) =  .25000    TCGLIRC( 3) =  6.9000    TCGLIRC( 4) =  9.1733     TCGLIRC( 5) =  10.601
TCGLIRC( 6) =  12.192    TCGLIRC( 7) =  13.926    TCGLIRC( 8) =  17.726    TCGLIRC( 9) = 0.          TCGLIRC(10) = 0.
TCGLIRC(11) = 0.         TCGLIRC(12) = 0.         TCGLIRC(13) = 0.         TCGLIRC(14) = 0.          TCGLIRC(15) = 0
TCGLIRC(16) = 0.         TCGLIRC(17) = 0.         TCGLIRC(18) = 0.         TCGLIRC(19) = 0.          TCGLIRC(20) = 0.
UVLLIRC( 1) = 0.         UVLLIRC( 2) = 0.         UVLLIRC( 3) = 0.
181 RL LIRC = 0.         182 RLOLIRC = 0.         183 VCGLIRC = 0.         184 PCGLIRC = 0.          185 CWTLIRC = 0.
186 TPELIRC =  2.0000    187 PVLLIRC = 0.         188 TLSLIRC = 0.         189 PHAPCRC =  1.0000     190 SW PCRC = 0
FLIPCRC( 1) = 0.         FLIPCRC( 2) = 0.         FLIPCRC( 3) = 0.
FDRPCRC( 1) = 0.         FDRPCRC( 2) = 0.         FDRPCRC( 3) = 0.
FMAPCRC( 1) = 0.         FMAPCRC( 2) = 0.         FMAPCRC( 3) = 0.
200 RM PCRC =  46.243    201 VOLPCRC =  .41413E+06 202 TLAPCRC = 0.        203 TLSPCRC = 0.          204 TDSPCRC = 0
205 DTIPCRC = 0.         206 TDUPCRC = 0.         207 TRFPCRC = 0.         208 OPFLAG  = 0.          209 FL GP   = 0.
210 FMTGP   = 0.
F11GP  ( 1) = -1.7406    F11GP  ( 2) =  .32582E-03 F11GP  ( 3) =  7.9108
T11GP  ( 1) = -2.8103    T11GP  ( 2) =  3.7499    T11GP  ( 3) = -.57450
FPPGP  ( 1) =  .30280E-08 FPPGP  ( 2) =  .82011E-08 FPPGP  ( 3) = -8.1000
TPPGP  ( 1) =  .35996E-08 TPPGP  ( 2) = -.13912E-08 TPPGP  ( 3) =  .61065E-09
223 TINGP   = 0.         224 TLAGP   = 0.
FSOGP  ( 1) = 0.         FSOGP  ( 2) = 0.         FSOGP  ( 3) = 0.
TSOGP  ( 1) = 0          TSOGP  ( 2) = 0.         TSOGP  ( 3) = 0.
FPOGP  ( 1) = 0.         FPOGP  ( 2) = 0.         FPOGP  ( 3) = 0.
TPOGP  ( 1) = 0.         TPOGP  ( 2) = 0.         TPOGP  ( 3) = 0
TRMGP  ( 1) =  821.00    TRMGP  ( 2) =  .28309E-09 TRMGP  ( 3) = -.19206E-09
240 DCFLAG  = 0.         241 FLALIDC = 0.         242 SWILIDC = 0.
FDOLIDC( 1) = 0.         FDOLIDC( 2) = 0.         FDOLIDC( 3) = 0.
TDOLIDC( 1) = 0.         TDOLIDC( 2) = 0.         TDOLIDC( 3) = 0.
FLPLIDC( 1) = 0.         FLPLIDC( 2) = 0.         FLPLIDC( 3) = 0.
FAPLIDC( 1) = 0.         FAPLIDC( 2) = 0.         FAPLIDC( 3) = 0.
VAPLIDC( 1) = 0          VAPLIDC( 2) = 0.         VAPLIDC( 3) = 0.
258 FLLLIDC = 0.         259 ELMLIDC = 0.         260 ELCLIDC = 0.         261 DEMLIDC = 0.          262 RMNLIDC = 0.
263 DISLIDC =  3.9783
CONLIDC( 1) = -1.0608    CONLIDC( 2) = 0.         CONLIDC( 3) =  .58100    CONLIDC( 4) = 0.
TCGLIDC( 1) = 0.         TCGLIDC( 2) =  .66065    TCGLIDC( 3) =  2.3226    TCGLIDC( 4) =  4.9961     TCGLIDC( 5) =  5.5187
TCGLIDC( 6) =  6.1781    TCGLIDC( 7) =  6.9735    TCGLIDC( 8) =  7.9052    TCGLIDC( 9) = 0.          TCGLIDC(10) = 0.
TCGLIDC(11) = 0.         TCGLIDC(12) = 0.         TCGLIDC(13) = 0.         TCGLIDC(14) = 0.          TCGLIDC(15) = 0.
TCGLIDC(16) = 0.         TCGLIDC(17) = 0.         TCGLIDC(18) = 0.         TCGLIDC(19) = 0.          TCGLIDC(20) = 0.
UVLLIDC( 1) = 0.         UVLLIDC( 2) = 0.         UVLLIDC( 3) = 0.
291 RL LIDC = 0.         292 RLOLIDC = 0.         293 VCGLIDC = 0.         294 PCGLIDC = 0.          295 CWTLIDC = 0.
296 TPELIDC =  1.0000    297 PVLLIDC = 0.         298 TLSLIDC = 0.         299 PHAPCDC =  1.0000     300 SW PCDC = 0.
FLIPCDC( 1) = 0.         FLIPCDC( 2) = 0.         FLIPCDC( 3) = 0.
FDRPCDC( 1) = 0.         FDRPCDC( 2) = 0.         FDRPCDC( 3) = 0.
FMAPCDC( 1) = 0.         FMAPCDC( 2) = 0.         FMAPCDC( 3) = 0.
310 RM PCDC =  12.496    311 VOLPCDC =  8172.7    312 TLAPCDC = 0.         313 TLSPCDC = 0.          314 TDSPCDC = 0
315 DTIPCDC = 0.         316 TDUPCDC = 0.         317 TRFPCDC = 0.
F21AS  ( 1) = 0.         F21AS  ( 2) = 0.         F21AS  ( 3) = 0.
```

Figure 8. (Continued)

```
T21AS  ( 1) = 0.          T21AS  ( 2) = 0.          T21AS  ( 3) = 0.
324 ALPAS   = 0.          325 BETAS   = 0.          326 VM AS   = 0.          327 Q  AS   = 0.          328 CX AS   = 0.
329 CY AS   = 0.          330 CZ AS   = 0.          331 CL AS   = 0.          332 CM AS   = 0.          333 CN AS   = 0.
334 EXLAS   = 0.          335 EXAAS   = 0.
CENAS  ( 1) = 0.          CENAS  ( 2) = 0.          CENAS  ( 3) = 0.
TCZAS  ( 1) = 0.          TCZAS  ( 2) =  .12500     TCZAS  ( 3) =  .28125     TCZAS  ( 4) =  .40675     TCZAS  ( 5) =  .56117
TCZAS  ( 6) =  2.3151     TCZAS  ( 7) =  2.4118     TCZAS  ( 8) =  2.5252     TCZAS  ( 9) =  2.5702     TCZAS  (10) = 0.
TCZAS  (11) = 0.          TCZAS  (12) = 0.          TCZAS  (13) = 0.          TCZAS  (14) = 0.          TCZAS  (15) = 0.
TCZAS  (16) = 0.          TCZAS  (17) = 0.          TCZAS  (18) = 0.          TCZAS  (19) = 0.          TCZAS  (20) = 0.
359 HD AS   =  2.8728     360 CEFLAG  = 0.
FPBRSCS( 1) = -58.330     FPBRSCS( 2) =  .11107E-01 FPBRSCS( 3) =  269.85
TPBRSCS( 1) =  5.3622     TPBRSCS( 2) = -135.69     TPBRSCS( 3) =  1.1854
FABRSCS( 1) =  58.318     FABRSCS( 2) = -.11215E-01 FABRSCS( 3) = -269.85
TABRSCS( 1) = -.38263E-01 TABRSCS( 2) =  4.5365     TABRSCS( 3) = -.86058E-02
TRMRSCS( 1) =  821.00     TRMRSCS( 2) =  .28309E-08 TRMRSCS( 3) = -.19206E-08
FPBRSKC( 1) = -5.4570     FPBRSKC( 2) =  .10317E-02 FPBRSKC( 3) =  24.807
TPBRSKC( 1) =  .38263E-01 TPBRSKC( 2) = -4.5365     TPBRSKC( 3) =  .86058E-02
FABRSKC( 1) =  5.4570     FABRSKC( 2) = -.10317E-02 FABRSKC( 3) = -24.807
TABRSKC( 1) =  .45140E-14 TABRSKC( 2) =  .37891E-10 TABRSKC( 3) =  .31449E-13
TRMRSKC( 1) =  821.00     TRMRSKC( 2) =  .56957E-08 TRMRSKC( 3) = -.55964E-08
391 GX CE   = -.84582E-08 392 GY CE   =  .24352E-08 393 GZ CE   = -.15178E-08 394 DR CE   =  .93770E-17
FADCE  ( 1) = 0.          FADCE  ( 2) = 0.          FADCE  ( 3) = 0.
TADCE  ( 1) = 0.          TADCE  ( 2) = 0.          TADCE  ( 3) = 0.
401 WT CE   =  250.70     402 S  CE   =  9.8500     403 B  CE   =  1.6200     404 C  CE   =  6.1000
CINCE  ( 1) =  18.970     CINCE  ( 2) =  18.920     CINCE  ( 3) =  2.4100     CINCE  ( 4) = -.46000
409 CX CE   = 0.          410 CY CE   = 0.          411 CZ CE   = 0.          412 CL CE   = 0.          413 CM CE   = 0
414 CN CE   = 0.          415 ALPCE   = 0.          416 BETCE   = 0.          417 VM CE   = 0.          418 Q  CE   = 0.
419 ALTCE   =  .65722     420 SEPCE   = 0.          421 GX SE   =  .17162E-07 422 GY SE   = -.76175E-08 423 GZ SE   =  .95567E-08
424 DR SE   = -.68549E-18 425 ALTSE   = -.52844     426 DREAM   =  .93770E-17 427 RADAM   =  .31741E-09 428 PTSAM   = 0.
429 PTIAM   = 0.

                                          PARAMETERS

  1 CTTIME  =  .10000E-02   2 SPTIME  =  .16600      3 MPTIME  =  1.7620      4 GPTIME  =  .18000      5 DCTIME  =  .15000
  6 CETIME  =  1.7620       7 H  AG   = 0.
WINAG  ( 1) = 0.          WINAG  ( 2) = 0.          WINAG  ( 3) = 0.
 11 BP AG   =  25.780      12 TE AG   =  73.800      13 SW AG   =  1.0000     14 UP CT   =  1.0000
SAPCT  ( 1) = -.65000     SAPCT  ( 2) = 0.          SAPCT  ( 3) = -3.2000
AAPCT  ( 1) = -.57180     AAPCT  ( 2) = 0.          AAPCT  ( 3) =  1.5930
 21 UCLCT   =  4.2208      22 CSKCT   =  2.8330      23 VI CT   =  18.500     24 PA CT   =  .78500     25 PT CT   =  10.000
 26 CBPCT   =  8000.0      27 C  CT   =  .17030E-02  28 CI CT   =  .10000E-04 29 PMWCT   =  25.500     30 SK CT   =  .10000E+06
 31 CK CT   =  150.00      32 GAMCT   =  1.1990      33 TF CT   =  2600.0     34 C1 CT   =  .12000     35 C2 CT   =  .30000
 36 B  CT   =  .33500E-01  37 BXPCT   =  .44000      38 TI CT   =  294.00     39 TDECT   = 0.
PCGSR  ( 1) = -.51000     PCGSR  ( 2) = 0.          PCGSR  ( 3) = -1.5600
EA SR  ( 1) = 0.          EA SR  ( 2) =  4.5300     EA SR  ( 3) = 0.
XRNSR  ( 1) =  .15100E-01 XRNSR  ( 2) = 0.          XRNSR  ( 3) =  1.5952
 49 YAWSR   = 0.           50 PITSR   = -53.780      51 PL SR   =  2.9170     52 PODSR   =  .21670     53 PIDSR   =  .14580
 54 AB WB   =  1.0000      55 SW WB   =  81.500
SX WB  ( 1) = -.27600E-01 SX WB  ( 2) =  .43100E-01 SX WB  ( 3) = -.60050
SM WB  ( 1) =  4.1497     SM WB  ( 2) =  4.4476     SM WB  ( 3) =  2.1539
SP WB  ( 1) =  .13110     SP WB  ( 2) =  1.1391     SP WB  ( 3) = -.38120
 65 W2 WB   = 0.
X2 WB  ( 1) = 0.          X2 WB  ( 2) = 0.          X2 WB  ( 3) = 0.
BM2WB  ( 1) = 0.          BM2WB  ( 2) = 0.          BM2WB  ( 3) = 0.
BP2WB  ( 1) = 0.          BP2WB  ( 2) = 0.          BP2WB  ( 3) = 0.
 75 W3 WB   = 0.
X3 WB  ( 1) = 0.          X3 WB  ( 2) = 0.          X3 WB  ( 3) = 0.
BM3WB  ( 1) = 0.          BM3WB  ( 2) = 0.          BM3WB  ( 3) = 0.
BP3WB  ( 1) = 0.          BP3WB  ( 2) = 0.          BP3WB  ( 3) = 0.
UD SL  ( 1) = 0.          UD SL  ( 2) = 0.          UD SL  ( 3) = 0.
WD SL  ( 1) = 0.          WD SL  ( 2) = 0.          WD SL  ( 3) = 0.
BL1RL  ( 1) = -.41250     BL1RL  ( 2) =  .66700     BL1RL  ( 3) =  .63330
BL2RL  ( 1) = -.47130     BL2RL  ( 2) =  .68700     BL2RL  ( 3) = -.11460
```

Figure 8. (Continued)

```
BL3RL   ( 1) = -.56110     BL3RL   ( 2) =  .68700     BL3RL   ( 3) = -1.2561
BL4RL   ( 1) = -.41250     BL4RL   ( 2) = -.66700     BL4RL   ( 3) =  .63330
BL5RL   ( 1) = -.47130     BL5RL   ( 2) = -.68700     BL5RL   ( 3) = -.11480
BL6RL   ( 1) = -.56110     BL6RL   ( 2) = -.66700     BL6RL   ( 3) = -1.2561
109 UP RL    =  1.0000     110 RLRRL    = -4.0410
XRRRL   ( 1) = -.57160     XRRRL   ( 2) =  .68700     XRRRL   ( 3) =  1.5930
114 RLLRL    = -4.0410
XRLRL   ( 1) = -.57160     XRLRL   ( 2) = -.66700     XRLRL   ( 3) =  1.5930
ERLRL   ( 1) = 0.          ERLRL   ( 2) =  17.000     ERLRL   ( 3) = 0.
SPRRL   ( 1) =  50000.     SPRRL   ( 2) =  50000.
DPGRL   ( 1) =  40.000     DPGRL   ( 2) =  40.000
125 SBFRL    =  .20000E-01 126 ZTSRL    = -3.0000     127 BTSRL    =  1.0000
CPTRL   ( 1) = 0.          CPTRL   ( 2) = 0.          CPTRL   ( 3) = 0.
131 YPRSP    =  2.0000     132 AVWSP    =  28650.     133 WMISP    =  .14700E-02 134 SMISP    =  .10000E-02 135 RIISP    =  .03600E-03
136 RIFSP    =  .51000E-03
XR SP   ( 1) =  .20840     XR SP   ( 2) = 0.          XR SP   ( 3) =  .70000
UV SP   ( 1) = 0           UV SP   ( 2) = 0.          UV SP   ( 3) = -1.9000
143 GSASP    = -30.000     144 GSFSP    =  30.000     145 SPRSP    =  5.0000     146 DPGSP    =  .15000E-01 147 FMTSP    =  800.00
148 TMXSP    =  .52000     149 TNFSP    =  .10400     150 TOSSP    = 0.          151 TSUSP    =  .50000E-02 152 GMASP    =  10.000
XYZMP   ( 1) = -.27770     XYZMP   ( 2) =  .17700E-01 XYZMP   ( 3) = -2.4725
EA MP   ( 1) = 0.          EA MP   ( 2) = 0.          EA MP   ( 3) = 0.
159 XR MP    =  8200.0     160 XD MP    =  80.000
ER MP   ( 1) =  750.00     ER MP   ( 2) =  750.00     ER MP   ( 3) =  750.00
ED MP   ( 1) =  2.0000     ED MP   ( 2) =  1.8000     ED MP   ( 3) =  2.0000
UV MP   ( 1) = -.60400E-01 UV MP   ( 2) =  .39900E-01 UV MP   ( 3) = -.99740
170 CSKMP    =  1.4583     171 VI MP    =  5.7500     172 PA MP    =  .51850     173 PT MP    =  10.000     174 CBPMP    =  9000.0
175 C  MP    =  .85150E-03 176 CI MP    =  .10000E-04 177 PMWMP    =  25.500     178 GAMMP    =  1.1990     179 TF MP    =  2600.0
180 C1 MP    =  .12000     181 C2 MP    =  .30000     182 B  MP    =  .33500E-01 183 BXPMP    =  .44000     184 TI MP    =  294.00
185 TDEMP    =  .50000E-01 186 OFFLIRC  = 0.          187 BLILIRC  =  1.0000
APXLIRC( 1)  = 0.          APXLIRC( 2)  = 0.          APXLIRC( 3)  = 0.
AP1LIRC( 1)  = 0.          AP1LIRC( 2)  = 0.          AP1LIRC( 3)  = -2.5900
AP2LIRC( 1)  = 0.          AP2LIRC( 2)  = 0.          AP2LIRC( 3)  = 0.
AP3LIRC( 1)  = 0.          AP3LIRC( 2)  = 0.          AP3LIRC( 3)  = 0.
AP4LIRC( 1)  = 0.          AP4LIRC( 2)  = 0.          AP4LIRC( 3)  = 0.
203 FTRLIRC  =  1.0000     204 FSOLIRC  =  10.000     205 ULLLIRC  =  550.00     206 ULSLIRC  =  .35000     207 GORLIRC  =  28.000
208 TYPLIRC  =  2.0000     209 STIPCRC  =  461.80     210 RSCPCRC  =  77.640     211 RFMPCRC  =  1.0000     212 RFDPCRC  =  1.1500
213 RFSPCRC  =  48.000     214 B  PCRC  =  .11000E-01 215 CI PCRC  =  [illegible].3300
CT PCRC( 1)  =  10.410     CT PCRC( 2)  =  88.100     CT PCRC( 3)  = -171.50
CN PCRC( 1)  = -38.200     CN PCRC( 2)  =  37.000     CN PCRC( 3)  =  39.700
CM PCRC( 1)  = 0.          CM PCRC( 2)  = 0.
224 FD PCRC  =  1.0000     225 PWTPCRC  =  20.190
PMIPCRC( 1)  =  .30520     PMIPCRC( 2)  =  .25770     PMIPCRC( 3)  =  .34900
PPIPCRC( 1)  =  .54600E-01 PPIPCRC( 2)  =  .14840     PPIPCRC( 3)  =  .97500E-01
232 TEMPCRC  =  .50000E-01 233 CSPPCRC  =  2000.0     234 CDPPCRC  =  14.000
DPGPCRC( 1)  =  2.0000     DPGPCRC( 2)  =  2.0000     DPGPCRC( 3)  =  2.0000
UV GP   ( 1) = -.61880     UV GP   ( 2) = -.34200     UV GP   ( 3) = -.70710
XMOGP   ( 1) = -.45000     XMOGP   ( 2) = -.20800     XMOGP   ( 3) = -1.3450
XYZGP   ( 1) = -.32000     XYZGP   ( 2) = -.33000     XYZGP   ( 3) = -.70000
EA GP   ( 1) = 0.          EA GP   ( 2) = 0.          EA GP   ( 3) = 0.
250 XR GP    =  2500.0     251 XD GP    =  30.000
ER GP   ( 1) =  500.00     ER GP   ( 2) =  500.00     ER GP   ( 3) =  500.00
ED GP   ( 1) =  .80000     ED GP   ( 2) =  .80000     ED GP   ( 3) =  .45000
258 TDEGP    = 0.          259 BLILIDC  =  2.0000
APXLIDC( 1)  = -3.8400     APXLIDC( 2)  = 0.          APXLIDC( 3)  = -1.0400
AP1LIDC( 1)  = -.55000     AP1LIDC( 2)  =  .51000     AP1LIDC( 3)  = -1.0400
AP2LIDC( 1)  = -.55000     AP2LIDC( 2)  = -.51000     AP2LIDC( 3)  = -1.0400
AP3LIDC( 1)  = 0.          AP3LIDC( 2)  = 0.          AP3LIDC( 3)  = 0.
AP4LIDC( 1)  = 0.          AP4LIDC( 2)  = 0.          AP4LIDC( 3)  = 0.
275 FTRLIDC  =  .50000     276 FSOLIDC  =  10.000     277 ULLLIDC  =  2030.0     278 ULSLIDC  =  .40000     279 GORLIDC  =  16.000
280 TYPLIDC  =  1.0000     281 STIPCDC  =  10.570     282 RSCPCDC  =  18.630     283 RFMPCDC  = 0.          284 RFDPCDC  = 0.
285 RFSPCDC  = 0.          286 B  PCDC  =  .80000E-01 287 CI PCDC  =  1.0000
CT PCDC( 1)  = -.54600     CT PCDC( 2)  = -24.400     CT PCDC( 3)  =  14.690
CN PCDC( 1)  =  31.200     CN PCDC( 2)  = -3.1100     CN PCDC( 3)  = -1.0700
CM PCDC( 1)  = -.64000     CM PCDC( 2)  =  .20000
296 FD PCDC  =  1.0000     297 PWTPCDC  =  8.1000
```

Figure 8. (Continued)

```
PMIPCDC( 1) =  .34000E-01 PMIPCDC( 2) =  .38000E-01 PMIPCDC( 3) =  .14000E-01
PPIPCDC( 1) = 0           PPIPCDC( 2) = 0.          PPIPCDC( 3) = 0.
304 TEMPCDC =  .50000E-01 305 CSPPCDC =  2000.0     306 CDPPCDC =  14.000
DPQPCDC( 1) =  2.0000     DPQPCDC( 2) =  2.0000     DPQPCDC( 3) =  2.0000
310 UP AS   =  1.0000     311 ZWSAS   = -3.5000
XEMAS  ( 1) = 0           XEMAS  ( 2) = 0.          XEMAS  ( 3) = -3.2500
315 CDXAS   = 0.          316 ECXAS   =  1.0000     317 ECYAS   =  1.0000     318 ECZAS   =  1.0000     319 CLPAS   = -.45380E-03
320 CMQAS   = -.10470E-01 321 CNRAS   = -.52360E-03 322 S  AS   =  6.9400
XYZRSCS( 1) =  .72040     XYZRSCS( 2) =  .19700E-01 XYZRSCS( 3) = -1.0636
EA RSCS( 1) = 0           EA RSCS( 2) = 0.          EA RSCS( 3) = 0.
329 XR RSCS =  27900.     330 XD RSCS =  300.00
ER RSCS( 1) =  1750.0     ER RSCS( 2) =  1750.0     ER RSCS( 3) =  1750.0
ED RSCS( 1) =  15.000     ED RSCS( 2) =  15.000     ED RSCS( 3) =  9.0000
337 FL RSKC = 0.
XYZRSKC( 1) = -.12160     XYZRSKC( 2) =  .16000E-02 XYZRSKC( 3) =  1.3841
EA RSKC( 1) = 0.          EA RSKC( 2) = 0.          EA RSKC( 3) = 0.
344 XR RSKC =  8200.0     345 XD RSKC =  120.00
ER RSKC( 1) =  750.00     ER RSKC( 2) =  750.00     ER RSKC( 3) =  750.00
ED RSKC( 1) =  1.8000     ED RSKC( 2) =  2.2000     ED RSKC( 3) =  1.8000
352 PC CE   =  95.000     353 CEWCE   =  49.800
CMICE  ( 1) =  10.368     CMICE  ( 2) =  10.516     CMICE  ( 3) =  3.0416
CPICE  ( 1) =  .97400E-01 CPICE  ( 2) =  2.0290     CPICE  ( 3) =  .31100E-01
360 CLPCE   = -.45400E-02 361 CMQCE   = -.10500E-01 362 CNRCE   = -.52400E-03
XSPCE  ( 1) = -.72040     XSPCE  ( 2) = -.19700E-01 XSPCE  ( 3) =  1.0636
366 WT ABSK =  25.400
BMIABSK( 1) =  .44090     BMIABSK( 2) =  .57740     BMIABSK( 3) =  .44010
BPIABSK( 1) =  .18410     BPIABSK( 2) =  .18600     BPIABSK( 3) =  .20790
FAUABSK( 1) = 0.          FAUABSK( 2) = 0.          FAUABSK( 3) = 0.
TAUABSK( 1) = 0.          TAUABSK( 2) = 0.          TAUABSK( 3) = 0.
F16SE  ( 1) = 0.          F16SE  ( 2) = 0.          F16SE  ( 3) = 0.
F17SE  ( 1) = 0.          F17SE  ( 2) = 0.          F17SE  ( 3) = 0.
F18SE  ( 1) = 0.          F18SE  ( 2) = 0.          F18SE  ( 3) = 0.
T16SE  ( 1) = 0.          T16SE  ( 2) = 0.          T16SE  ( 3) = 0.
T17SE  ( 1) = 0.          T17SE  ( 2) = 0.          T17SE  ( 3) = 0.
T18SE  ( 1) = 0.          T18SE  ( 2) = 0.          T18SE  ( 3) = 0.
F25SE  ( 1) = 0.          F25SE  ( 2) = 0.          F25SE  ( 3) = 0.
F26SE  ( 1) = 0.          F26SE  ( 2) = 0.          F26SE  ( 3) = 0.
F27SE  ( 1) = 0.          F27SE  ( 2) = 0.          F27SE  ( 3) = 0.
F28SE  ( 1) = 0.          F28SE  ( 2) = 0.          F28SE  ( 3) = 0.
T25SE  ( 1) = 0           T25SE  ( 2) = 0.          T25SE  ( 3) = 0.
T26SE  ( 1) = 0           T26SE  ( 2) = 0.          T26SE  ( 3) = 0.
T27SE  ( 1) = 0.          T27SE  ( 2) = 0.          T27SE  ( 3) = 0.
T28SE  ( 1) = 0           T28SE  ( 2) = 0.          T28SE  ( 3) = 0.
421 FL AM   =  1.0000     422 PRTAM   = 0.          423 EXPAM   =  2.0000     424 GXPAM   =  35.000     425 GXNAM   =  30.000
426 GYLAM   =  15.000     427 GZLAM   =  12.000     428 DRPAM   =  18.000     429 DRNAM   =  16.000     430 RDLAM   =  1.0000


                          SYSTEM EIGENVALUES AT THIS OPERATING POINT

                                      76  EIGENVALUES
MODE      REAL          IMAGINARY       NATURAL FREQ.     DAMPING RATIO
  1     -6.83441      +-  38.7573          39.3552          .173659
  2     -8.11328      +- 647.927          647.977           .125209E-01
  3     -8.31389      +-  41.8308          42.6490          .194938
  4    -11.1231       +- 367.031          367.199           .302916E-01
  5    -11.5821       +-  68.0732          69.0514          .167731
  6    -11.8496       +-  51.5558          52.9000          .224000
  7    -11.8496       +-  51.5558          52.9000          .224000
  8    -14.4361       +-  47.5446          49.6879          .290535
  9    -15.0524       +- 307.478          307.846           .488958E-01
 10    -16.8548       +-  69.2130          71.2357          .236606
 11    -44.5805       +- 124.611          132.346           .336849
```

Figure 8. (Continued)

```
12    -57.2183      +-  80.0081           88.3624         .581689
13    -58.5337      +-  80.2772           89.3510         .589161
14    -58.7314      +-  80.2774           89.4678         .590456
15    -68.4918      +-  71.4005           98.8402         .692255
16    -68.5246      +-  77.3080          103.306          .663316
17    -68.8038      +-  76.5988          102.829          .887163
18    -83.2050      +-  50.5096           97.3360         .854823
19    -91.0988      +-  250.081          288.157          .342274
20    -93.2898      +-  234.273          252.168          .369990
21    -98.1237      +-  44.8788          107.900          .909397
22    -101.326      +-  48.2748          111.393          .909630
23    -103.134      +-  342.813          357.991          .288090
24    -105.136      +-  207.850          232.928          .451369
25    -120.727      +-  377.335          396.178          .304731
26    -156.206      +-  203.922          258.875          .608102
27    -204.448      +-  128.881          241.680          .845944
28    -225.218      +-  116.726          253.669          .887842
29    -225.218      +-  116.726          253.669          .887842
30    -225.218      +-  116.726          253.669          .887842
31    -225.218      +-  116.726          253.669          .887842
32    -225.218      +-  116.726          253.669          .887842
33    -225.218      +-  116.726          253.669          .887842
34    -296.873      +-  124.064          321.846          .922718
35    -588.066      +-  335.798          677.188          .868397
36    -609.321      +-  623.012          871.444          .699208
37    -681.803      +-  819.289          920.924          .740129
38    -928.397      +-  1093.97         1434.82           .647050


          7.19300     CPU SECONDS WERE REQUIRED FOR THE PREVIOUS ANALYSIS

COMMAND CARD --->     XIC-X

                                    I.C./OPERATING POINT

  1 EF CT    = 0.           2 EL CT    = 0.            3 WK CT    = 0.           4 WB CT    = 0.          5 WI SR   =  5.6000
UAPSL  ( 1) =  821.00     UAPSL  ( 2) = 0.          UAPSL  ( 3) = 0.
XAPSL  ( 1) = 0.          XAPSL  ( 2) = 0.          XAPSL  ( 3) = 0.
WAPSL  ( 1) = 0.          WAPSL  ( 2) = 0.          WAPSL  ( 3) = 0.
EAPSL  ( 1) = 0.          EAPSL  ( 2) = 0.          EAPSL  ( 3) = 0.
 18 WG SP   = 0.
ESGSP  ( 1) = 0.          ESGSP  ( 2) = -12.380     ESGSP  ( 3) = 0.
ESRSP  ( 1) = 0.          ESRSP  ( 2) = -14.000     ESRSP  ( 3) = 0.
 25 EF MP   = 0.           26 EL MP   = 0.           27 WK MP   = 0.          28 WB MP   = 0.         29 EC LIRC = 0.
 30 TF LIRC = 0.
UPPFCRC( 1) =  821.00     UPPPCRC( 2) =  .22426E-16 UPPPCRC( 3) =  .17685E-18
XPPPCRC( 1) = -1.2191     XPPPCRC( 2) =  .17788E-01 XPPPCRC( 3) = -1.3254
WPPPCRC( 1) = 0.          WPPPCRC( 2) = 0.          WPPPCRC( 3) = 0.
EPPPCRC( 1) =  .73738E-03 EPPPCRC( 2) =  12.409     EPPPCRC( 3) =  .23599E-02
UPCPCRC( 1) =  821.00     UPCPCRC( 2) = -.10115E-16 UPCPCRC( 3) =  .26379E-19
XPCPCRC( 1) = -1.2191     XPCPCRC( 2) =  .17788E-01 XPCPCRC( 3) = -1.8249
 49 EC LIDC = 0.           50 TF LIDC = 0.
UPPPCDC( 1) =  821.00     UPPPCDC( 2) = -.45449E-20 UPPPCDC( 3) =  .22589E-19
XPPPCDC( 1) = -.87956     XPPPCDC( 2) = -.32998     XPPPCDC( 3) = -.85219E-01
WPPPCDC( 1) = -.48101E-19 WPPPCDC( 2) = -.61689E-18 WPPPCDC( 3) = -.22183E-18
EPPPCDC( 1) =  .73738E-03 EPPPCDC( 2) =  12.409     EPPPCDC( 3) =  .23599E-02
UPCPCDC( 1) =  821.00     UPCPCDC( 2) =  .67369E-20 UPCPCDC( 3) =  .10738E-19
XPCPCDC( 1) = -.87956     XPCPCDC( 2) = -.32998     XPCPCDC( 3) = -.81719E-01
```

Figure 8. (Continued)

```
                                   /*/*/*/   SIMULATION ANALYSIS   /*/*/*/

PRATE =   1   OUTRATE =  10   PRINT CONTROL =   5   INT MODE =   2   TINC =   .10000E-02   TMAX =   5.0000
PRATE2 =  10   OUTRATE2 =  10   PRINT2 =   5   PRINT2 FROM  .30000     TO  5.0000       TINC2 =   .10000E-02

                                      EASIEST EJECTION SEAT MODEL

          CASE NO.   2                         80/12/11.   03.42.03.

       TIME = 0.
UDST     (1)   = -.53907E-06 UDST     (2)   =  .27377E-06 UDST     (3)   = -.30816E-06 USTSE    (1)   =  801.82
USTSE    (2)   = -.32895E-02 USTSE    (3)   =  176.42     SRPSE    (1)   = -.41862     SRPSE    (2)   = -.33568E-05
SRPSE    (3)   =  .52844     WDST     (1)   =  .97802E-05 WDST     (2)   =  .21802E-04 WDST     (3)   =  .10385E-04
WSTSE    (1)   = -.31381E-31 WSTSE    (2)   =  .80908E-30 WSTSE    (3)   =  .70327E-31 ESTSE    (1)   =  .73738E-03
ESTSE    (2)   =  12.409     ESTSE    (3)   =  .23589E-02 RADAM          =  .31741E-09 DREAM          =  .93770E-17
CF CT          =  0.         UCPCE    (1)   =  801.83     UCPCE    (2)   = -.33010E-02 UCPCE    (3)   =  176.39
XCPCE    (1)   =  .58399E-01 XCPCE    (2)   =  .19747E-01 XCPCE    (3)   = -.65722     ECPCE    (1)   =  .74230E-03
ECPCE    (2)   =  12.406     ECPCE    (3)   =  .23828E-02 XPCPCRC(1)    = -1.2191     XPCPCRC(2)    =  .17788E-01
XPCPCRC(3)     = -1.8249     XPCPCDC(1)    = -.87956     XPCPCDC(2)    = -.32998     XPCPCDC(3)    = -.84719E-01

    CATAPULT IGNITION AT TIME =       .0014 SEC


       TIME =  .1000E-01
UDST     (1)   =  3.4681     UDST     (2)   =  .48835     UDST     (3)   = -23.068     USTSE    (1)   =  801.82
USTSE    (2)   = -.43290E-02 USTSE    (3)   =  176.25     SRPSE    (1)   =  7.7932     SRPSE    (2)   = -.29823E-05
SRPSE    (3)   =  .52602     WDST     (1)   = -.85297     WDST     (2)   = -40.837     WDST     (3)   = -7.4247
WSTSE    (1)   = -.98946E-02 WSTSE    (2)   = -1.0545     WSTSE    (3)   = -.35754E-01 ESTSE    (1)   =  .88441E-03
ESTSE    (2)   =  12.403     ESTSE    (3)   =  .24980E-02 RADAM          =  .13757E-02 DREAM          =  .53977E-02
CF CT          =  127.42     UCPCE    (1)   =  801.83     UCPCE    (2)   = -.48328E-02 UCPCE    (3)   =  176.35
XCPCE    (1)   =  8.2684     XCPCE    (2)   =  .19748E-01 XCPCE    (3)   = -.65727     ECPCE    (1)   =  .88077E-03
ECPCE    (2)   =  12.405     ECPCE    (3)   =  .24670E-02 XPCPCRC(1)    =  6.9909     XPCPCRC(2)    =  .17790E-01
XPCPCRC(3)     = -1.8250     XPCPCDC(1)    =  7.3304     XPCPCDC(2)    = -.32898     XPCPCDC(3)    = -.84817E-01

       TIME =  .2000E-01
UDST     (1)   =  2.8387     UDST     (2)   =  .43106     UDST     (3)   = -25.837     USTSE    (1)   =  801.86
USTSE    (2)   = -.34184E-02 USTSE    (3)   =  176.00     SRPSE    (1)   =  18.003     SRPSE    (2)   = -.56104E-05
SRPSE    (3)   =  .52477     WDST     (1)   = -10.240     WDST     (2)   = -20.544     WDST     (3)   = -11.878
WSTSE    (1)   =  .18957E-01 WSTSE    (2)   = -1.3283     WSTSE    (3)   = -.68187E-02 ESTSE    (1)   =  .77063E-03
ESTSE    (2)   =  12.391     ESTSE    (3)   =  .26498E-02 RADAM          =  .27295E-02 DREAM          =  .41170E-01
CF CT          =  305.19     UCPCE    (1)   =  801.86     UCPCE    (2)   = -.75190E-03 UCPCE    (3)   =  176.18
XCPCE    (1)   =  18.478     XCPCE    (2)   =  .19749E-01 XCPCE    (3)   = -.65765     ECPCE    (1)   =  .56431E-03
ECPCE    (2)   =  12.396     ECPCE    (3)   =  .23404E-02 XPCPCRC(1)    =  15.201     XPCPCRC(2)    =  .17796E-01
XPCPCRC(3)     = -1.8260     XPCPCDC(1)    =  15.540     XPCPCDC(2)    = -.32998     XPCPCDC(3)    = -.85694E-01

       TIME =  .3000E-01
UDST     (1)   =  .18007E-01 UDST     (2)   = -.75657     UDST     (3)   = -44.814     USTSE    (1)   =  801.89
USTSE    (2)   = -.92314E-04 USTSE    (3)   =  175.73     SRPSE    (1)   =  24.212     SRPSE    (2)   = -.48263E-05
SRPSE    (3)   =  .52283     WDST     (1)   =  31.907     WDST     (2)   = -246.26     WDST     (3)   =  44.524
WSTSE    (1)   =  .46907E-01 WSTSE    (2)   = -1.4806     WSTSE    (3)   =  .20820E-01 ESTSE    (1)   =  .51219E-03
ESTSE    (2)   =  12.377     ESTSE    (3)   =  .26203E-02 RADAM          =  .72894E-02 DREAM          =  .11533
CF CT          =  548.05     UCPCE    (1)   =  801.91     UCPCE    (2)   =  .22441E-02 UCPCE    (3)   =  175.89
XCPCE    (1)   =  24.688     XCPCE    (2)   =  .19750E-01 XCPCE    (3)   = -.65808     ECPCE    (1)   =  .35099E-03
ECPCE    (2)   =  12.381     ECPCE    (3)   =  .23601E-02 XPCPCRC(1)    =  23.411     XPCPCRC(2)    =  .17801E-01
XPCPCRC(3)     = -1.8280     XPCPCDC(1)    =  23.750     XPCPCDC(2)    = -.32998     XPCPCDC(3)    = -.87616E-01

       TIME =  .4000E-01
UDST     (1)   =  11.107     UDST     (2)   =  .64256     UDST     (3)   = -150.14     USTSE    (1)   =  801.92
USTSE    (2)   = -.52565E-02 USTSE    (3)   =  174.76     SRPSE    (1)   =  32.421     SRPSE    (2)   = -.97740E-05
SRPSE    (3)   =  .51852     WDST     (1)   = -4.3400     WDST     (2)   = -509.09     WDST     (3)   = -17.115
WSTSE    (1)   =  .46575E-01 WSTSE    (2)   = -5.9324     WSTSE    (3)   = -.19225E-01 ESTSE    (1)   =  .12884E-02
ESTSE    (2)   =  12.342     ESTSE    (3)   =  .36959E-02 RADAM          =  .10053E-01 DREAM          =  .23676
CF CT          =  855.58     UCPCE    (1)   =  801.98     UCPCE    (2)   = -.49424E-02 UCPCE    (3)   =  175.41
```

Figure 9. Simulation Output

```
XCPCE  (1)    =  32.898    XCPCE  (2)    =  .19754E-01 XCPCE  (3)    = -.66077     ECPCE  (1)    =  .10745E-02
ECPCE  (2)    =  12.359    ECPCE  (3)    =  .30872E-02 XPCPCRC(1)    =  31.821     XPCPCRC(2)    =  .17814E-01
XPCPCRC(3)    = -1.8313    XPCPCDC(1)    =  31.959     XPCPCDC(2)    = -.32998     XPCPCDC(3)    = -.90802E-01

       TIME =  .5000E-01
UDST   (1)    =  23.503    UDST   (2)    =  1.1602     UDST   (3)    = -248.43     USTSE  (1)    =  802.10
USTSE  (2)    = -.63908E-02 USTSE (3)    =  172.76     SRPSE  (1)    =  40.627     SRPSE  (2)    = -.12950E-04
SRPSE  (3)    =  .50798    WDST   (1)    = -9.1853     WDST   (2)    = -502.45     WDST   (3)    = -31.193
WSTSE  (1)    =  .17843    WSTSE  (2)    = -11.102     WSTSE  (3)    = -.20478E-01 ESTSE  (1)    =  .13488E-02
ESTSE  (2)    =  12.257    ESTSE  (3)    =  .49571E-02 RADAM         =  .33477E-01 DREAM         =  .51248
CF CT         =  1228.0    UCPCE  (1)    =  802.16     UCPCE  (2)    =  .42369E-02 UCPCE  (3)    =  174.24
XCPCE  (1)    =  41.107    XCPCE  (2)    =  .18769E-01 XCPCE  (3)    = -.68547     ECPCE  (1)    =  .55620E-03
ECPCE  (2)    =  12.303    ECPCE  (3)    =  .34172E-02 XPCPCRC(1)    =  39.831     XPCPCRC(2)    =  .17858E-01
XPCPCRC(3)    = -1.8395    XPCPCDC(1)    =  40.187     XPCPCDC(2)    = -.32997     XPCPCDC(3)    = -.98407E-01

       TIME =  .6000E-01
UDST   (1)    =  35.103    UDST   (2)    =  1.8464     UDST   (3)    = -320.54     USTSE  (1)    =  802.40
USTSE  (2)    =  .15170E-01 USTSE (3)    =  169.80     SRPSE  (1)    =  48.830     SRPSE  (2)    = -.19759E-04
SRPSE  (3)    =  .48858    WDST   (1)    =  25.635     WDST   (2)    = -268.99     WDST   (3)    =  19.074
WSTSE  (1)    =  .17890    WSTSE  (2)    = -15.019     WSTSE  (3)    = -.13133     ESTSE  (1)    =  .10181E-03
ESTSE  (2)    =  12.124    ESTSE  (3)    =  .81048E-02 RADAM         =  .68684E-01 DREAM         =  1.0849
CF CT         =  1619.2    UCPCE  (1)    =  802.51     UCPCE  (2)    =  .28554E-01 UCPCE  (3)    =  171.95
XCPCE  (1)    =  49.315    XCPCE  (2)    =  .19798E-01 XCPCE  (3)    = -.87564     ECPCE  (1)    = -.10682E-02
ECPCE  (2)    =  12.192    ECPCE  (3)    =  .38154E-02 XPCPCRC(1)    =  48.040     XPCPCRC(2)    =  .17925E-01
XPCPCRC(3)    = -1.8571    XPCPCDC(1)    =  48.373     XPCPCDC(2)    = -.32995     XPCPCDC(3)    = -.11500

       TIME =  .7000E-01
UDST   (1)    =  36.912    UDST   (2)    =  2.8011     UDST   (3)    = -385.24     USTSE  (1)    =  802.76
USTSE  (2)    =  .33548E-01 USTSE (3)    =  166.43     SRPSE  (1)    =  57.030     SRPSE  (2)    = -.50355E-04
SRPSE  (3)    =  .45855    WDST   (1)    = -10.907     WDST   (2)    = -51.034     WDST   (3)    = -8.5772
WSTSE  (1)    =  .28298    WSTSE  (2)    = -18.666     WSTSE  (3)    = -.94613E-01 ESTSE  (1)    = -.64246E-03
ESTSE  (2)    =  11.964    ESTSE  (3)    =  .87922E-02 RADAM         =  .12638     DREAM         =  2.0154
CF CT         =  2020.1    UCPCE  (1)    =  802.98     UCPCE  (2)    =  .46718E-01 UCPCE  (3)    =  168.54
XCPCE  (1)    =  57.521    XCPCE  (2)    =  .19805E-01 XCPCE  (3)    = -.69465     ECPCE  (1)    = -.19658E-02
ECPCE  (2)    =  12.026    ECPCE  (3)    =  .56428E-02 XPCPCRC(1)    =  56.248     XPCPCRC(2)    =  .18008E-01
XPCPCRC(3)    = -1.8887    XPCPCDC(1)    =  58.575     XPCPCDC(2)    = -.32994     XPCPCDC(3)    = -.14351

       TIME =  .8000E-01
UDST   (1)    =  32.413    UDST   (2)    =  .28678     UDST   (3)    = -355.77     USTSE  (1)    =  803.12
USTSE  (2)    =  .46252E-01 USTSE (3)    =  162.78     SRPSE  (1)    =  65.227     SRPSE  (2)    = -.60432E-04
SRPSE  (3)    =  .41604    WDST   (1)    =  19.029     WDST   (2)    =  297.52     WDST   (3)    =  11.428
WSTSE  (1)    =  .38255    WSTSE  (2)    = -15.270     WSTSE  (3)    =  .27083E-01 ESTSE  (1)    = -.11138E-02
ESTSE  (2)    =  11.801    ESTSE  (3)    =  .11653E-01 RADAM         =  .20288     DREAM         =  3.2430
CF CT         =  2351.7    UCPCE  (1)    =  803.46     UCPCE  (2)    =  .51532E-01 UCPCE  (3)    =  164.40
XCPCE  (1)    =  65.724    XCPCE  (2)    =  .19831E-01 XCPCE  (3)    = -.72588     ECPCE  (1)    = -.14641E-02
ECPCE  (2)    =  11.832    ECPCE  (3)    =  .95548E-02 XPCPCRC(1)    =  64.453     XPCPCRC(2)    =  .18126E-01
XPCPCRC(3)    = -1.9295    XPCPCDC(1)    =  64.774     XPCPCDC(2)    = -.32992     XPCPCDC(3)    = -.18534

       TIME =  .9000E-01
UDST   (1)    =  18.255    UDST   (2)    =  1.3351     UDST   (3)    = -324.11     USTSE  (1)    =  803.38
USTSE  (2)    =  .57982E-01 USTSE (3)    =  159.37     SRPSE  (1)    =  73.419     SRPSE  (2)    = -.76927E-04
SRPSE  (3)    =  .35977    WDST   (1)    = -10.185     WDST   (2)    =  421.94     WDST   (3)    =  7.9672
WSTSE  (1)    =  .29820    WSTSE  (2)    = -11.521     WSTSE  (3)    = -.32628E-01 ESTSE  (1)    = -.10252E-02
ESTSE  (2)    =  11.668    ESTSE  (3)    =  .15350E-01 RADAM         =  .28782     DREAM         =  4.6022
CF CT         =  2654.5    UCPCE  (1)    =  803.81     UCPCE  (2)    =  .45650E-01 UCPCE  (3)    =  160.20
XCPCE  (1)    =  73.922    XCPCE  (2)    =  .19870E-01 XCPCE  (3)    = -.77227     ECPCE  (1)    =  .93726E-04
ECPCE  (2)    =  11.654    ECPCE  (3)    =  .15130E-01 XPCPCRC(1)    =  72.652     XPCPCRC(2)    =  .18271E-01
XPCPCRC(3)    = -1.9863    XPCPCDC(1)    =  72.969     XPCPCDC(2)    = -.32989     XPCPCDC(3)    = -.24141

       TIME =  .1000E+00
UDST   (1)    =  3.0754    UDST   (2)    =  .23071     UDST   (3)    = -310.32     USTSE  (1)    =  803.48
USTSE  (2)    =  .56048E-01 USTSE (3)    =  156.24     SRPSE  (1)    =  81.606     SRPSE  (2)    = -.87446E-04
SRPSE  (3)    =  .28774    WDST   (1)    = -10.889     WDST   (2)    =  243.09     WDST   (3)    = -19.578
WSTSE  (1)    =  .33658    WSTSE  (2)    = -8.0472     WSTSE  (3)    =  .74825E-01 ESTSE  (1)    = -.48989E-03
ESTSE  (2)    =  11.570    ESTSE  (3)    =  .18663E-01 RADAM         =  .36830     DREAM         =  5.8876
```

Figure 9. (Continued)

```
CF CT           = 2883.1      UCPCE  (1)    =  803.92      UCPCE  (2)    =  .47231E-01 UCPCE  (3)    =  156.48
XCPCE  (1)      = 82.115      XCPCE  (2)    =  .19922E-01  XCPCE  (3)    = -.83610     ECPCE  (1)    =  .10383E-02
ECPCE  (2)      = 11.527      ECPCE  (3)    =  .20644E-01  XPCPCRC(1)    =  80.845     XPCPCRC(2)    =  .18422E-01
XPCPCRC(3)      = -2.0578     XPCPCDC(1)    =  81.158      XPCPCDC(2)    = -.32986     XPCPCDC(3)    = -.31255

        TIME =  .1100
UDST   (1)      = -5.2800     UDST   (2)    =  .18857E-02  UDST   (3)    = -308.54     USTSE  (1)    =  803.47
USTSE  (2)      =  .65909E-01 USTSE  (3)    =  153.14      SRPSE  (1)    =  89.788     SRPSE  (2)    = -.84013E-04
SRPSE  (3)      =  .18687     WDST   (1)    = -1.2184      WDST   (2)    =  154.54     WDST   (3)    =  21.886
WSTSE  (1)      =  .14580     WSTSE  (2)    = -5.8758      WSTSE  (3)    = -.21895E-01 ESTSE  (1)    = -.87343E-03
ESTSE  (2)      =  11.501     ESTSE  (3)    =  .20944E-01  RADAM         =  .43587     DREAM         =  6.9642
CF CT           =  3305.1     UCPCE  (1)    =  803.82      UCPCE  (2)    =  .58486E-01 UCPCE  (3)    =  153.39
XCPCE  (1)      =  90.302     XCPCE  (2)    =  .19980E-01  XCPCE  (3)    = -.91947     ECPCE  (1)    =  .91891E-03
ECPCE  (2)      =  11.481     ECPCE  (3)    =  .24875E-01  XPCPCRC(1)    =  89.031     XPCPCRC(2)    =  .18549E-01
XPCPCRC(3)      = -2.1487     XPCPCDC(1)    =  89.342      XPCPCDC(2)    = -.32983     XPCPCDC(3)    = -.40101

        TIME =  .1200
UDST   (1)      = -11.805     UDST   (2)    =  1.3708      UDST   (3)    = -320.16     USTSE  (1)    =  803.38
USTSE  (2)      =  .68291E-01 USTSE  (3)    =  150.03      SRPSE  (1)    =  97.964     SRPSE  (2)    = -.94589E-04
SRPSE  (3)      =  .84450E-01 WDST   (1)    = -15.870      WDST   (2)    =  5.1216     WDST   (3)    = -8.0387
WSTSE  (1)      =  .51821E-01 WSTSE  (2)    = -5.0178      WSTSE  (3)    = -.66857E-01 ESTSE  (1)    = -.71788E-03
ESTSE  (2)      =  11.447     ESTSE  (3)    =  .22145E-01  RADAM         =  .48895     DREAM         =  7.8089
CF CT           =  3603.1     UCPCE  (1)    =  803.58      UCPCE  (2)    =  .78717E-01 UCPCE  (3)    =  150.61
XCPCE  (1)      =  98.481     XCPCE  (2)    =  .20027E-01  XCPCE  (3)    = -1.0246     ECPCE  (1)    = -.50413E-03
ECPCE  (2)      =  11.434     ECPCE  (3)    =  .25853E-01  XPCPCRC(1)    =  97.209     XPCPCRC(2)    =  .18622E-01
XPCPCRC(3)      = -2.2585     XPCPCDC(1)    =  97.519      XPCPCDC(2)    = -.32980     XPCPCDC(3)    = -.51041

        TIME =  .1300
UDST   (1)      = -13.180     UDST   (2)    = -.76183      UDST   (3)    = -361.58     USTSE  (1)    =  803.25
USTSE  (2)      =  .71898E-01 USTSE  (3)    =  148.84      SRPSE  (1)    =  108.13     SRPSE  (2)    = -.92758E-04
SRPSE  (3)      = -.51952E-01 WDST   (1)    =  7.7188      WDST   (2)    = -165.66     WDST   (3)    =  15.546
WSTSE  (1)      = -.15372E-01 WSTSE  (2)    = -5.8997      WSTSE  (3)    =  .24847E-01 ESTSE  (1)    = -.11585E-02
ESTSE  (2)      =  11.394     ESTSE  (3)    =  .22031E-01  RADAM         =  .53019     DREAM         =  8.4652
CF CT           =  3922.3     UCPCE  (1)    =  803.33      UCPCE  (2)    =  .92325E-01 UCPCE  (3)    =  147.78
XCPCE  (1)      =  108.65     XCPCE  (2)    =  .20050E-01  XCPCE  (3)    = -1.1537     ECPCE  (1)    = -.19479E-02
ECPCE  (2)      =  11.417     ECPCE  (3)    =  .24658E-01  XPCPCRC(1)    =  105.38     XPCPCRC(2)    =  .18629E-01
XPCPCRC(3)      = -2.3903     XPCPCDC(1)    =  105.69      XPCPCDC(2)    = -.32980     XPCPCDC(3)    = -.64387

        TIME =  .1400
UDST   (1)      = -7.1126     UDST   (2)    =  .94722      UDST   (3)    = -392.64     USTSE  (1)    =  803.13
USTSE  (2)      =  .71227E-01 USTSE  (3)    =  142.77      SRPSE  (1)    =  114.29     SRPSE  (2)    = -.96623E-04
SRPSE  (3)      = -.21512     WDST   (1)    = -25.197      WDST   (2)    =  88.357     WDST   (3)    = -38.195
WSTSE  (1)      = -.57156E-01 WSTSE  (2)    = -7.5883      WSTSE  (3)    = -.66966E-01 ESTSE  (1)    = -.10938E-02
ESTSE  (2)      =  11.328     ESTSE  (3)    =  .21886E-01  RADAM         =  .56405     DREAM         =  9.0190
CF CT           =  4182.3     UCPCE  (1)    =  803.13      UCPCE  (2)    =  .86338E-01 UCPCE  (3)    =  144.41
XCPCE  (1)      =  114.81     XCPCE  (2)    =  .20048E-01  XCPCE  (3)    = -1.3091     ECPCE  (1)    = -.21523E-02
ECPCE  (2)      =  11.382     ECPCE  (3)    =  .22488E-01  XPCPCRC(1)    =  113.54     XPCPCRC(2)    =  .18585E-01
XPCPCRC(3)      = -2.5509     XPCPCDC(1)    =  113.85      XPCPCDC(2)    = -.32981     XPCPCDC(3)    = -.80423

        TIME =  .1500
UDST   (1)      = -21.421     UDST   (2)    =  .98128      UDST   (3)    = -162.21     USTSE  (1)    =  803.08
USTSE  (2)      =  .89422E-01 USTSE  (3)    =  139.95      SRPSE  (1)    =  122.44     SRPSE  (2)    = -.99262E-04
SRPSE  (3)      = -.40420     WDST   (1)    =  36.473      WDST   (2)    =  1533.8     WDST   (3)    =  42.590
WSTSE  (1)      = -.81123E-02 WSTSE  (2)    =  2.3105      WSTSE  (3)    = -.13002     ESTSE  (1)    = -.31188E-02
ESTSE  (2)      =  11.287     ESTSE  (3)    =  .20335E-01  RADAM         =  .59547     DREAM         =  9.5213
CF CT           =  3819.7     UCPCE  (1)    =  802.99      UCPCE  (2)    =  .91101E-01 UCPCE  (3)    =  140.59
XCPCE  (1)      =  122.97     XCPCE  (2)    =  .20026E-01  XCPCE  (3)    = -1.4929     ECPCE  (1)    = -.32782E-02
ECPCE  (2)      =  11.324     ECPCE  (3)    =  .19731E-01  XPCPCRC(1)    =  121.70     XPCPCRC(2)    =  .18552E-01
XPCPCRC(3)      = -2.7405     XPCPCDC(1)    =  122.00      XPCPCDC(2)    = -.32982     XPCPCDC(3)    = -.99353

        TIME =  .1600
UDST   (1)      = -54.861     UDST   (2)    = -.55380      UDST   (3)    =  48.235     USTSE  (1)    =  802.68
USTSE  (2)      =  .99381E-01 USTSE  (3)    =  139.42      SRPSE  (1)    =  130.59     SRPSE  (2)    = -.11785E-03
SRPSE  (3)      = -.81274     WDST   (1)    =  12.713      WDST   (2)    =  1407.1     WDST   (3)    =  18.653
WSTSE  (1)      = -.10405     WSTSE  (2)    =  16.685      WSTSE  (3)    =  .21083E-01 ESTSE  (1)    = -.33748E-02
```

Figure 9. (Continued)

```
ESTSE  (2)    =  11.384    ESTSE  (3)    =  .20023E-01 RADAM        =  .80907     DREAM        =  9.7430
CF CT         =  3479.2    UCPCE  (1)    =  802.64     UCPCE  (2)   =  .64306E-01 UCPCE  (3)   =  137.77
XCPCE  (1)    =  131.11    XCPCE  (2)    =  .19909E-01 XCPCE  (3)   = -1.7040     ECPCE  (1)   = -.14181E-02
ECPCE  (2)    =  11.321    ECPCE  (3)    =  .20274E-01 XPCPCRC(1)   =  129.84     XPCPCRC(2)   =  .18496E-01
XPCPCRC(3)    = -2.8531    XPCPCDC(1)    =  130.15     XPCPCDC(2)   = -.32904     XPCPCDC(3)   = -1.2069

    PITCH STAPAC IGNITION AT TIME=     .1870  SEC

        TIME =  .1700
UDST   (1)    = -43.512    UDST   (2)    = -2.2870     UDST   (3)   =  111.70     USTSE  (1)   =  802.05
USTSE  (2)    =  .85751E-01 USTSE  (3)   =  140.55     SRPSE  (1)   =  130.73     SRPSE  (2)   = -.12228E-03
SRPSE  (3)    = -.83968    WDST   (1)    =  106.51     WDST   (2)   =  2501.5     WDST   (3)   = -76.352
WSTSE  (1)    =  .41050    WSTSE  (2)    =  31.155     WSTSE  (3)   =  .19663     ESTSE  (1)   = -.11789E-02
ESTSE  (2)    =  11.815    ESTSE  (3)    =  .21388E-01 RADAM        =  .58555     DREAM        =  9.3087
CF CT         =  3163.9    UCPCE  (1)    =  801.88     UCPCE  (2)   =  .82030E-02 UCPCE  (3)   =  137.38
XCPCE  (1)    =  139.25    XCPCE  (2)    =  .19845E-01 XCPCE  (3)   = -1.9411     ECPCE  (1)   =  .33875E-02
ECPCE  (2)    =  11.466    ECPCE  (3)    =  .24182E-01 XPCPCRC(1)   =  137.97     XPCPCRC(2)   =  .18472E-01
XPCPCRC(3)    = -3.1809    XPCPCDC(1)    =  138.29     XPCPCDC(2)   = -.32986     XPCPCDC(3)   = -1.4385

        TIME =  .1800
UDST   (1)    = -150.16    UDST   (2)    = -4.0581     UDST   (3)   =  147.07     USTSE  (1)   =  800.69
USTSE  (2)    =  .43653E-01 USTSE  (3)   =  142.24     SRPSE  (1)   =  146.87     SRPSE  (2)   = -.82500E-04
SRPSE  (3)    = -1.0938    WDST   (1)    = -145.22     WDST   (2)   = -1288.8     WDST   (3)   = -223.38
WSTSE  (1)    = -.42691    WSTSE  (2)    =  28.288     WSTSE  (3)   =  .29188     ESTSE  (1)   =  .21350E-02
ESTSE  (2)    =  11.857    ESTSE  (3)    =  .24528E-01 RADAM        =  .52847     DREAM        =  8.4376
CF CT         =  2885.5    UCPCE  (1)    =  800.69     UCPCE  (2)   = -.23848E-01 UCPCE  (3)   =  139.57
XCPCE  (1)    =  147.38    XCPCE  (2)    =  .19858E-01 XCPCE  (3)   = -2.2004     ECPCE  (1)   =  .71877E-02
ECPCE  (2)    =  11.789    ECPCE  (3)    =  .30143E-01 XPCPCRC(1)   =  146.10     XPCPCRC(2)   =  .185??E-01
XPCPCRC(3)    = -3.4258    XPCPCDC(1)    =  146.42     XPCPCDC(2)   = -.32985     XPCPCDC(3)   = -1.6832

    GUN      IGNITION AT TIME =      .1810 SEC

    GUN      STRIPOFF AT TIME =      .1831 SEC

    DRAG     CHUTE LAUNCH AT TIME =      .1831 SEC

        TIME =  .1900
UDST   (1)    = -228.11    UDST   (2)    = -70.769     UDST   (3)   = -235.65     USTSE  (1)   =  799.93
USTSE  (2)    = -.30019    USTSE  (3)    =  142.66     SRPSE  (1)   =  155.00     SRPSE  (2)   =  .10222E-02
SRPSE  (3)    = -1.3689    WDST   (1)    = -547.36     WDST   (2)   = -1863.8     WDST   (3)   = -628.45
WSTSE  (1)    =  11.295    WSTSE  (2)    =  3.5501     WSTSE  (3)   =  4.5083     ESTSE  (1)   =  .12609E-01
ESTSE  (2)    =  12.122    ESTSE  (3)    =  .80801E-01 RADAM        =  .46708     DREAM        =  7.3590
CF CT         =  2592.5    UCPCE  (1)    =  799.69     UCPCE  (2)   =  .73043E-01 UCPCE  (3)   =  142.78
XCPCE  (1)    =  155.50    XCPCE  (2)    =  .20575E-01 XCPCE  (3)   = -2.4818     ECPCE  (1)   =  .10421E-01
ECPCE  (2)    =  12.177    ECPCE  (3)    =  .45931E-01 XPCPCRC(1)   =  154.21     XPCPCRC(2)   =  .20152E-01
XPCPCRC(3)    = -3.8862    XPCPCDC(1)    =  154.41     XPCPCDC(2)   = -.38178     XPCPCDC(3)   = -2.0584

        TIME =  .2000
UDST   (1)    =  8.2918    UDST   (2)    =  82.149     UDST   (3)   =  210.54     USTSE  (1)   =  799.58
USTSE  (2)    =  .89288    USTSE  (3)    =  141.89     SRPSE  (1)   =  163.11     SRPSE  (2)   =  .21624E-03
SRPSE  (3)    = -1.8634    WDST   (1)    =  869.94     WDST   (2)   =  3344.4     WDST   (3)   =  810.81
WSTSE  (1)    = -1.4485    WSTSE  (2)    =  27.786     WSTSE  (3)   = -6.4967     ESTSE  (1)   = -.17928E-01
ESTSE  (2)    =  12.232    ESTSE  (3)    =  .10605     RADAM        =  .40433     DREAM        =  8.4253
CF CT         =  2339.4    UCPCE  (1)    =  798.90     UCPCE  (2)   =  .25296     UCPCE  (3)   =  143.57
XCPCE  (1)    =  163.61    XCPCE  (2)    =  .21489E-01 XCPCE  (3)   = -2.7028     ECPCE  (1)   =  .51958E-02
ECPCE  (2)    =  12.384    ECPCE  (3)    =  .98337E-01 XPCPCRC(1)   =  162.33     XPCPCRC(2)   =  .22927E-01
XPCPCRC(3)    = -3.9903    XPCPCDC(1)    =  162.33     XPCPCDC(2)   = -.48054     XPCPCDC(3)   = -2.5187

        TIME =  .2100
UDST   (1)    = -183.76    UDST   (2)    =  30.676     UDST   (3)   =  450.02     USTSE  (1)   =  798.36
USTSE  (2)    =  .73317    USTSE  (3)    =  145.58     SRPSE  (1)   =  171.22     SRPSE  (2)   =  .31460E-03
```

Figure 9. (Continued)

```
SRPSE   (3)    = -1.9784    WDST    (1)    = -598.78    WDST    (2)    =  799.64    WDST    (3)    = -493.02
WSTSE   (1)    =  4.2475    WSTSE   (2)    =  42.198    WSTSE   (3)    = -.33351    ESTSE   (1)    = -.33865E-01
ESTSE   (2)    =  12.804    ESTSE   (3)    =  .13395    RADAM          =  .35842    DREAM          =  5.7120
CF CT          =  2113.2    UCPCE   (1)    =  798.07    UCPCE   (2)    =  1.0260    UCPCE   (3)    =  144.22
XCPCE   (1)    =  171.72    XCPCE   (2)    =  .22282E-01 XCPCE  (3)    = -3.1030    ECPCE   (1)    = -.45354E-01
ECPCE   (2)    =  12.577    ECPCE   (3)    =  .12412    XPCPCRC(1)     =  170.43    XPCPCRC(2)     =  .24399E-01
XPCPCRC(3)     = -4.3054    XPCPCDC(1)     =  170.24    XPCPCDC(2)     = -.51487    XPCPCDC(3)     = -3.0017

     CATAPULT STRIPOFF AT TIME =      .2155 SEC

     SUSTAINER ROCKET ON AT TIME =      .2185 SEC

     END OF GUIDED STROKE AT TIME =   .2188 SEC

          TIME =  .2200
UDST    (1)    = -10.934    UDST    (2)    =  82.176    UDST    (3)    =  1325.2    USTSE   (1)    =  798.95
USTSE   (2)    =  1.1178    USTSE   (3)    =  152.84    SRPSE   (1)    =  179.34    SRPSE   (2)    =  .30826E-03
SRPSE   (3)    = -2.3060    WDST    (1)    = -712.78    WDST    (2)    =  4775.8    WDST    (3)    = -452.09
WSTSE   (1)    =  2.2571    WSTSE   (2)    =  109.30    WSTSE   (3)    = -1.4020    ESTSE   (1)    = -.42381E-01
ESTSE   (2)    =  13.218    ESTSE   (3)    =  .15978    RADAM          =  .28843    DREAM          =  5.0338
CF CT          = 0.         UCPCE   (1)    =  798.85    UCPCE   (2)    =  1.1245    UCPCE   (3)    =  147.58
XCPCE   (1)    =  178.82    XCPCE   (2)    =  .22806E-01 XCPCE  (3)    = -3.4421    ECPCE   (1)    = -.43891E-01
ECPCE   (2)    =  12.848    ECPCE   (3)    =  .17185    XPCPCRC(1)     =  178.51    XPCPCRC(2)     =  .25838E-01
XPCPCRC(3)     = -4.6319    XPCPCDC(1)     =  178.08    XPCPCDC(2)     = -.53886    XPCPCDC(3)     = -3.4376

          TIME =  .2300
UDST    (1)    = -553.80    UDST    (2)    =  24.275    UDST    (3)    =  478.32    USTSE   (1)    =  793.37
USTSE   (2)    =  1.1985    USTSE   (3)    =  163.12    SRPSE   (1)    =  187.45    SRPSE   (2)    =  .16003E-02
SRPSE   (3)    = -2.6501    WDST    (1)    =  211.65    WDST    (2)    = -4846.7    WDST    (3)    = -1031.8
WSTSE   (1)    =  .31745    WSTSE   (2)    =  50.548    WSTSE   (3)    = -2.8978    ESTSE   (1)    = -.41008E-01
ESTSE   (2)    =  14.085    ESTSE   (3)    =  .17418    RADAM          =  .27728    DREAM          =  4.1898
CF CT          = 0.         UCPCE   (1)    =  794.33    UCPCE   (2)    =  1.1247    UCPCE   (3)    =  157.10
XCPCE   (1)    =  187.91    XCPCE   (2)    =  .23710E-01 XCPCE  (3)    = -3.7971    ECPCE   (1)    = -.32174E-01
ECPCE   (2)    =  13.752    ECPCE   (3)    =  .20785    XPCPCRC(1)     =  186.59    XPCPCRC(2)     =  .27314E-01
XPCPCRC(3)     = -4.8638    XPCPCDC(1)     =  185.82    XPCPCDC(2)     = -.53453    XPCPCDC(3)     = -3.8089

     SEAT/RAIL SEPARATION AT TIME   .2399 SEC

          TIME =  .2400
UDST    (1)    = -465.33    UDST    (2)    = -10.098    UDST    (3)    =  578.91    USTSE   (1)    =  790.82
USTSE   (2)    =  1.3289    USTSE   (3)    =  166.21    SRPSE   (1)    =  195.53    SRPSE   (2)    =  .16477E-02
SRPSE   (3)    = -3.0075    WDST    (1)    = -64.227    WDST    (2)    = -10528.    WDST    (3)    =  1751.9
WSTSE   (1)    =  6.7657    WSTSE   (2)    =  50.894    WSTSE   (3)    =  2.5244    ESTSE   (1)    = -.53014E-01
ESTSE   (2)    =  14.806    ESTSE   (3)    =  .22975    RADAM          =  .26761    DREAM          =  3.8423
CF CT          = 0.         UCPCE   (1)    =  790.95    UCPCE   (2)    =  1.4945    UCPCE   (3)    =  167.37
XCPCE   (1)    =  195.99    XCPCE   (2)    =  .24728E-01 XCPCE  (3)    = -4.1855    ECPCE   (1)    = -.53381E-01
ECPCE   (2)    =  14.805    ECPCE   (3)    =  .21844    XPCPCRC(1)     =  194.66    XPCPCRC(2)     =  .29459E-01
XPCPCRC(3)     = -5.3081    XPCPCDC(1)     =  193.41    XPCPCDC(2)     = -.49824    XPCPCDC(3)     = -4.1262

          TIME =  .2500
UDST    (1)    = -123.80    UDST    (2)    =  101.70    UDST    (3)    = -117.42    USTSE   (1)    =  788.60
USTSE   (2)    =  1.6174    USTSE   (3)    =  188.92    SRPSE   (1)    =  203.80    SRPSE   (2)    =  .10772E-02
SRPSE   (3)    = -3.3800    WDST    (1)    =  882.97    WDST    (2)    = -2512.2    WDST    (3)    =  22.050
WSTSE   (1)    =  1.0511    WSTSE   (2)    =  5.7360    WSTSE   (3)    = -6.9047    ESTSE   (1)    = -.56342E-01
ESTSE   (2)    =  14.792    ESTSE   (3)    =  .27364    RADAM          =  .31420    DREAM          =  3.6982
CF CT          = 0.         UCPCE   (1)    =  789.06    UCPCE   (2)    =  1.8107    UCPCE   (3)    =  174.45
XCPCE   (1)    =  204.06    XCPCE   (2)    =  .25784E-01 XCPCE  (3)    = -4.5457    ECPCE   (1)    = -.51227E-01
ECPCE   (2)    =  15.220    ECPCE   (3)    =  .26434    XPCPCRC(1)     =  202.71    XPCPCRC(2)     =  .31949E-01
XPCPCRC(3)     = -5.6745    XPCPCDC(1)     =  200.84    XPCPCDC(2)     = -.43510    XPCPCDC(3)     = -4.4108

          TIME =  .2600
```

Figure 9. (Continued)

```
UDST    (1)    = -37.422    UDST    (2)    =  89.002       UDST    (3)    = -527.38       USTSE   (1)    =  787.83
USTSE   (2)    =  2.1577    USTSE   (3)    =  185.82       SRPSE   (1)    =  211.65       SRPSE   (2)    =  .67549E-03
SRPSE   (3)    = -3.7883    WDST    (1)    = -1343.0       WDST    (2)    = -2054.2       WDST    (3)    = -868.41
WSTSE   (1)    =  8.0304    WSTSE   (2)    = -24.821       WSTSE   (3)    = -3.9843       ESTSE   (1)    = -.87714E-01
ESTSE   (2)    =  14.803    ESTSE   (3)    =  .33533       RADAM          =  .34125       DREAM          =  4.3807
CF CT          = 0.         UCPCE   (1)    =  786.83       UCPCE   (2)    =  2.2736       UCPCE   (3)    =  174.71
XCPCE   (1)    =  212.12    XCPCE   (2)    =  .26450E-01   XCPCE   (3)    = -4.8384       ECPCE   (1)    = -.83587E-01
ECPCE   (2)    =  15.338    ECPCE   (3)    =  .29434       XPCPCRC(1)     =  210.76       XPCPCRC(2)     =  .33886E-01
XPCPCRC(3)     = -8.0821    XPCPCDC(1)     =  208.12       XPCPCDC(2)     = -.35315       XPCPCDC(3)     = -4.8840

        TIME =  .2700
UDST    (1)    = -12.280    UDST    (2)    =  76.968       UDST    (3)    = -968.07       USTSE   (1)    =  787.58
USTSE   (2)    =  3.0888    USTSE   (3)    =  158.71       SRPSE   (1)    =  219.68       SRPSE   (2)    =  .85173E-03
SRPSE   (3)    = -4.1717    WDST    (1)    =  1610.6       WDST    (2)    = -5258.1       WDST    (3)    =  640.50
WSTSE   (1)    =  4.5880    WSTSE   (2)    = -87.068       WSTSE   (3)    = -8.1045       ESTSE   (1)    = -.14880
ESTSE   (2)    =  14.322    ESTSE   (3)    =  .34931       RADAM          =  .38208       DREAM          =  8.3343
CF CT          = 0.         UCPCE   (1)    =  786.71       UCPCE   (2)    =  3.0843       UCPCE   (3)    =  188.24
XCPCE   (1)    =  220.18    XCPCE   (2)    =  .26880E-01   XCPCE   (3)    = -5.3377       ECPCE   (1)    = -.14845
ECPCE   (2)    =  14.870    ECPCE   (3)    =  .33347       XPCPCRC(1)     =  218.80       XPCPCRC(2)     =  .36105E-01
XPCPCRC(3)     = -8.4713    XPCPCDC(1)     =  215.23       XPCPCDC(2)     = -.25777       XPCPCDC(3)     = -4.8577

        TIME =  .2800
UDST    (1)    =  86.371    UDST    (2)    =  171.80       UDST    (3)    = -1529.5       USTSE   (1)    =  788.00
USTSE   (2)    =  4.1438    USTSE   (3)    =  145.95       SRPSE   (1)    =  227.60       SRPSE   (2)    =  .76171E-03
SRPSE   (3)    = -4.5889    WDST    (1)    = -780.71       WDST    (2)    = -3405.1       WDST    (3)    = -87.828
WSTSE   (1)    =  3.7292    WSTSE   (2)    = -98.318       WSTSE   (3)    = -10.813       ESTSE   (1)    = -.22050
ESTSE   (2)    =  13.527    ESTSE   (3)    =  .40753       RADAM          =  .43073       DREAM          =  8.2530
CF CT          = 0.         UCPCE   (1)    =  788.01       UCPCE   (2)    =  4.0230       UCPCE   (3)    =  155.82
XCPCE   (1)    =  228.18    XCPCE   (2)    =  .27046E-01   XCPCE   (3)    = -5.7495       ECPCE   (1)    = -.21385
ECPCE   (2)    =  14.182    ECPCE   (3)    =  .37801       XPCPCRC(1)     =  226.85       XPCPCRC(2)     =  .38703E-01
XPCPCRC(3)     = -8.9005    XPCPCDC(1)     =  222.20       XPCPCDC(2)     = -.15390       XPCPCDC(3)     = -5.2389

        TIME =  .2900
UDST    (1)    =  138.63    UDST    (2)    =  105.77       UDST    (3)    = -2182.4       USTSE   (1)    =  789.04
USTSE   (2)    =  6.3011    USTSE   (3)    =  127.65       SRPSE   (1)    =  235.68       SRPSE   (2)    =  .10883E-02
SRPSE   (3)    = -5.0183    WDST    (1)    =  894.26       WDST    (2)    = -4602.4       WDST    (3)    = -339.44
WSTSE   (1)    =  8.4142    WSTSE   (2)    = -145.43       WSTSE   (3)    = -7.6410       ESTSE   (1)    = -.31205
ESTSE   (2)    =  12.328    ESTSE   (3)    =  .43585       RADAM          =  .46555       DREAM          =  6.8454
CF CT          = 0.         UCPCE   (1)    =  790.16       UCPCE   (2)    =  5.4231       UCPCE   (3)    =  137.67
XCPCE   (1)    =  236.20    XCPCE   (2)    =  .26818E-01   XCPCE   (3)    = -6.1713       ECPCE   (1)    = -.31837
ECPCE   (2)    =  12.934    ECPCE   (3)    =  .41379       XPCPCRC(1)     =  234.89       XPCPCRC(2)     =  .41722E-01
XPCPCRC(3)     = -7.3491    XPCPCDC(1)     =  228.03       XPCPCDC(2)     = -.44924E-01   XPCPCDC(3)     = -5.5315

        TIME =  .3000
UDST    (1)    =  163.83    UDST    (2)    =  217.69       UDST    (3)    = -2652.2       USTSE   (1)    =  790.87
USTSE   (2)    =  7.1834    USTSE   (3)    =  103.43       SRPSE   (1)    =  243.66       SRPSE   (2)    =  .95845E-03
SRPSE   (3)    = -5.4830    WDST    (1)    =  433.12       WDST    (2)    = -3959.3       WDST    (3)    =  1129.8
WSTSE   (1)    =  4.8814    WSTSE   (2)    = -180.56       WSTSE   (3)    = -15.067       ESTSE   (1)    = -.45060
ESTSE   (2)    =  10.894    ESTSE   (3)    =  .48409       RADAM          =  .46892       DREAM          =  8.9870
CF CT          = 0.         UCPCE   (1)    =  792.78       UCPCE   (2)    =  8.9247       UCPCE   (3)    =  113.71
XCPCE   (1)    =  244.21    XCPCE   (2)    =  .26588E-01   XCPCE   (3)    = -8.6023       ECPCE   (1)    = -.43443
ECPCE   (2)    =  11.283    ECPCE   (3)    =  .46358       XPCPCRC(1)     =  242.93       XPCPCRC(2)     =  .45019E-01
XPCPCRC(3)     = -7.8158    XPCPCDC(1)     =  235.71       XPCPCDC(2)     =  .86897E-01   XPCPCDC(3)     = -5.8376

    DRAG      CHUTE LINESTRETCH AT TIME =       .3855 SEC

        TIME =  .4000
UDST    (1)    = -1730.5    UDST    (2)    =  684.81       UDST    (3)    = -4011.8       USTSE   (1)    =  730.71
USTSE   (2)    =  45.398    USTSE   (3)    = -292.34       SRPSE   (1)    =  322.68       SRPSE   (2)    =  .16283E-01
SRPSE   (3)    = -10.520    WDST    (1)    = -84.553       WDST    (2)    =  1655.2       WDST    (3)    = -762.75
WSTSE   (1)    =  22.882    WSTSE   (2)    = -310.38       WSTSE   (3)    = -61.824       ESTSE   (1)    = -4.1623
ESTSE   (2)    = -17.821    ESTSE   (3)    =  2.4270       RADAM          =  .17054       DREAM          =  2.4410
CF CT          = 0.         UCPCE   (1)    =  736.70       UCPCE   (2)    =  44.723       UCPCE   (3)    = -288.45
XCPCE   (1)    =  323.72    XCPCE   (2)    =  .53651E-02   XCPCE   (3)    = -11.284       ECPCE   (1)    = -4.1318
```

Figure 9. (Continued)

```
ECPCE  (2)     = -17.617    ECPCE  (3)     =  2.4177    XPCPCRC(1)     =  323.18         XPCPCRC(2)     =  .10180
XPCPCRC(3)     = -12.953    XPCPCDC(1)     =  303.58    XPCPCDC(2)     =  .77885         XPCPCDC(3)     = -10.356

    DRAG     CANOPY FILLED AT TIME =      .4350 SEC

         TIME =  .5000
UDST   (1)     = -2393.6    UDST   (2)     =  1806.2    UDST   (3)     =  6215.8         USTSE  (1)     =  507.81
USTSE  (2)     =  640.03    USTSE  (3)     = -58.793    SRPSE  (1)     =  309.40         SRPSE  (2)     =  .23677
SRPSE  (3)     = -15.591    WDST   (1)     =  4035.2    WDST   (2)     =  724.69         WDST   (3)     =  20358.
WSTSE  (1)     = -254.04    WSTSE  (2)     =  440.58    WSTSE  (3)     = -274.21         ESTSE  (1)     = -47.207
ESTSE  (2)     = -3.5280    ESTSE  (3)     = -2.3882    RADAM          =  2.1146         DREAM          = -.54757
CF CT          = 0.         UCPCE  (1)     =  494.17    UCPCE  (2)     =  538.54         UCPCE  (3)     = -58.342
XCPCE  (1)     =  400.11    XCPCE  (2)     = -.91240    XCPCE  (3)     = -16.602         ECPCE  (1)     = -47.700
ECPCE  (2)     = -3.8819    ECPCE  (3)     = -3.3000    XPCPCRC(1)     =  399.34         XPCPCRC(2)     =  .28420
XPCPCRC(3)     = -18.088    XPCPCDC(1)     =  378.51    XPCPCDC(2)     =  .59463E-01     XPCPCDC(3)     = -15.303

    PITCH STAPAC BURNOUT AT TIME=     .5175  SEC

    SUSTAINER ROCKET OFF AT TIME=     .5670  SEC

         TIME =  .6000
UDST   (1)     =  338.38    UDST   (2)     = -3622.8    UDST   (3)     = -2043.7         USTSE  (1)     =  643.86
USTSE  (2)     =  133.58    USTSE  (3)     =  77.455    SRPSE  (1)     =  469.16         SRPSE  (2)     = -2.2217
SRPSE  (3)     = -20.457    WDST   (1)     = -2334.4    WDST   (2)     = -7071.1         WDST   (3)     = -11545.
WSTSE  (1)     =  504.79    WSTSE  (2)     = -64.740    WSTSE  (3)     =  386.11         ESTSE  (1)     = -8.8346
ESTSE  (2)     =  18.215    ESTSE  (3)     =  32.004    RADAM          =  .70727         DREAM          =  1.0319
CF CT          = 0.         UCPCE  (1)     =  643.89    UCPCE  (2)     =  145.41         UCPCE  (3)     =  91.103
XCPCE  (1)     =  469.89    XCPCE  (2)     = -1.6762    XCPCE  (3)     = -21.556         ECPCE  (1)     = -7.9561
ECPCE  (2)     =  18.844    ECPCE  (3)     =  31.847    XPCPCRC(1)     =  468.58         XPCPCRC(2)     = -.78510
XPCPCRC(3)     = -22.387    XPCPCDC(1)     =  446.53    XPCPCDC(2)     = -1.1354         XPCPCDC(3)     = -19.768

         TIME =  .7000
UDST   (1)     = -3833.1    UDST   (2)     =  3767.7    UDST   (3)     = -2142.2         USTSE  (1)     =  484.65
USTSE  (2)     =  295.48    USTSE  (3)     = -219.36    SRPSE  (1)     =  532.34         SRPSE  (2)     = -4.2795
SRPSE  (3)     = -24.251    WDST   (1)     = -37.377    WDST   (2)     =  5750.9         WDST   (3)     =  332.16
WSTSE  (1)     =  285.68    WSTSE  (2)     = -110.29    WSTSE  (3)     = -601.27         ESTSE  (1)     = -34.304
ESTSE  (2)     =  22.459    ESTSE  (3)     =  63.938    RADAM          =  .86624         DREAM          = -8.9867
CF CT          = 0.         UCPCE  (1)     =  488.71    UCPCE  (2)     =  282.68         UCPCE  (3)     = -217.67
XCPCE  (1)     =  533.46    XCPCE  (2)     = -3.7880    XCPCE  (3)     = -24.859         ECPCE  (1)     = -34.314
ECPCE  (2)     =  22.455    ECPCE  (3)     =  63.954    XPCPCRC(1)     =  533.11         XPCPCRC(2)     = -2.0496
XPCPCRC(3)     = -25.149    XPCPCDC(1)     =  510.37    XPCPCDC(2)     = -2.8506         XPCPCDC(3)     = -23.976

         TIME =  .8000
UDST   (1)     =  2225.2    UDST   (2)     = -2342.9    UDST   (3)     =  1283.9         USTSE  (1)     =  384.11
USTSE  (2)     =  304.47    USTSE  (3)     = -271.04    SRPSE  (1)     =  590.57         SRPSE  (2)     = -8.3563
SRPSE  (3)     = -27.389    WDST   (1)     = -56.312    WDST   (2)     =  3312.3         WDST   (3)     =  3218.8
WSTSE  (1)     =  280.40    WSTSE  (2)     =  392.14    WSTSE  (3)     =  113.98         ESTSE  (1)     = -21.972
ESTSE  (2)     =  46.900    ESTSE  (3)     =  103.72    RADAM          =  .70642         DREAM          = -4.9549
CF CT          = 0.         UCPCE  (1)     =  373.98    UCPCE  (2)     =  311.95         UCPCE  (3)     = -278.62
XCPCE  (1)     =  591.73    XCPCE  (2)     = -5.8721    XCPCE  (3)     = -27.737         ECPCE  (1)     = -22.408
ECPCE  (2)     =  47.163    ECPCE  (3)     =  103.46    XPCPCRC(1)     =  591.78         XPCPCRC(2)     = -4.2265
XPCPCRC(3)     = -28.770    XPCPCDC(1)     =  569.05    XPCPCDC(2)     = -4.3146         XPCPCDC(3)     = -26.193

         TIME =  .9000
UDST   (1)     = -1160.9    UDST   (2)     =  1265.3    UDST   (3)     =  2858.4         USTSE  (1)     =  482.90
USTSE  (2)     =  205.02    USTSE  (3)     =  17.238    SRPSE  (1)     =  644.76         SRPSE  (2)     = -8.0588
SRPSE  (3)     = -30.582    WDST   (1)     =  764.70    WDST   (2)     = -3057.4         WDST   (3)     = -3170.4
WSTSE  (1)     =  247.03    WSTSE  (2)     =  451.18    WSTSE  (3)     = -161.84         ESTSE  (1)     =  19.337
ESTSE  (2)     =  13.108    ESTSE  (3)     =  151.13    RADAM          =  .54989         DREAM          =  2.4754
CF CT          = 0.         UCPCE  (1)     =  475.03    UCPCE  (2)     =  206.50         UCPCE  (3)     =  15.617
XCPCE  (1)     =  645.58    XCPCE  (2)     = -7.2752    XCPCE  (3)     = -29.815         ECPCE  (1)     =  18.458
ECPCE  (2)     =  12.656    ECPCE  (3)     =  150.93    XPCPCRC(1)     =  644.61         XPCPCRC(2)     = -6.8584
```

Figure 9. (Continued)

XPCPCRC(3)	= -28.383	XPCPCDC(1)	= 622.77	XPCPCDC(2)	= -6.3628	XPCPCDC(3)	= -28.754
TIME =	1.000						
UDST (1)	= 199.37	UDST (2)	= -687.88	UDST (3)	= -2584.6	USTSE (1)	= 367.66
USTSE (2)	= 301.92	USTSE (3)	= 32.108	SRPSE (1)	= 694.82	SRPSE (2)	= -9.1965
SRPSE (3)	= -33.095	WDST (1)	= 3815.5	WDST (2)	= -3933.2	WDST (3)	= 4976.3
WSTSE (1)	= 593.22	WSTSE (2)	= 108.79	WSTSE (3)	= 105.85	ESTSE (1)	= 38.487
ESTSE (2)	= -10.961	ESTSE (3)	= 181.72	RADAM	= .83932	DREAM	= 4.6307
CF CT	= 0.	UCPCE (1)	= 364.52	UCPCE (2)	= 314.87	UCPCE (3)	= 37.268
XCPCE (1)	= 895.48	XCPCE (2)	= -9.8873	XCPCE (3)	= -31.924	ECPCE (1)	= 36.488
ECPCE (2)	= -11.501	ECPCE (3)	= 191.20	XPCPCRC(1)	= 694.57	XPCPCRC(2)	= -10.081
XPCPCRC(3)	= -30.783	XPCPCDC(1)	= 672.84	XPCPCDC(2)	= -8.5816	XPCPCDC(3)	= -31.138
TIME =	1.100						
UDST (1)	= -449.35	UDST (2)	= -2337.7	UDST (3)	= -901.67	USTSE (1)	= 398.85
USTSE (2)	= 48.040	USTSE (3)	= -208.54	SRPSE (1)	= 740.92	SRPSE (2)	= -10.129
SRPSE (3)	= -34.465	WDST (1)	= -2384.9	WDST (2)	= 970.98	WDST (3)	= -4589.5
WSTSE (1)	= 557.04	WSTSE (2)	= -71.648	WSTSE ( )	= 52.874	ESTSE (1)	= 28.269
ESTSE (2)	= 1.5441	ESTSE (3)	= 257.54	RADAM	= .27417	DREAM	= -.30945
CF CT	= 0.	UCPCE (1)	= 397.65	UCPCE (2)	= 53.148	UCPCE (3)	= -206.33
XCPCE (1)	= 742.12	XCPCE (2)	= -10.745	XCPCE (3)	= -34.289	ECPCE (1)	= 28.377
ECPCE (2)	= 1.7293	ECPCE (3)	= 257.75	XPCPCRC(1)	= 741.79	XPCPCRC(2)	= -12.428
XPCPCRC(3)	= -33.939	XPCPCDC(1)	= 719.68	XPCPCDC(2)	= -10.321	XPCPCDC(3)	= -32.956
TIME =	1.200						
UDST (1)	= -622.74	UDST (2)	= -2525.0	UDST (3)	= 1964.8	USTSE (1)	= 363.21
USTSE (2)	= -173.93	USTSE (3)	= -148.76	SRPSE (1)	= 784.77	SRPSE (2)	= -11.808
SRPSE (3)	= -35.430	WDST (1)	= 611.83	WDST (2)	= -824.51	WDST (3)	= 1354.2
WSTSE (1)	= 801.78	WSTSE (2)	= 23.208	WSTSE (3)	= 159.13	ESTSE (1)	= 29.2[illegible]4
ESTSE (2)	= 5.5882	ESTSE (3)	= 312.40	RADAM	= .23449	DREAM	= .12[illegible]81
CF CT	= 0.	UCPCE (1)	= 362.57	UCPCE (2)	= -160.40	UCPCE (3)	= -149.60
XCPCE (1)	= 785.80	XCPCE (2)	= -12.170	XCPCE (3)	= -36.239	ECPCE (1)	= 29.270
ECPCE (2)	= 5.5271	ECPCE (3)	= 312.38	XPCPCRC(1)	= 785.31	XPCPCRC(2)	= -13.615
XPCPCRC(3)	= -37.093	XPCPCDC(1)	= 783.35	XPCPCDC(2)	= -11.946	XPCPCDC(3)	= -35.400
TIME =	1.300						
UDST (1)	= 241.72	UDST (2)	= 1005.8	UDST (3)	= 935.73	USTSE (1)	= 311.57
USTSE (2)	= -248.95	USTSE (3)	= 46.402	SRPSE (1)	= 826.40	SRPSE (2)	= -13.900
SRPSE (3)	= -38.446	WDST (1)	= -2913.1	WDST (2)	= -2151.7	WDST (3)	= -3160.0
WSTSE (1)	= 419.85	WSTSE (2)	= -158.39	WSTSE (3)	= -77.443	ESTSE (1)	= 38.363
ESTSE (2)	= 4.3887	ESTSE (3)	= 387.78	RADAM	= .58673	DREAM	= .78725
CF CT	= 0.	UCPCE (1)	= 313.95	UCPCE (2)	= -242.02	UCPCE (3)	= 52.152
XCPCE (1)	= 826.89	XCPCE (2)	= -13.388	XCPCE (3)	= -37.570	ECPCE (1)	= 38.479
ECPCE (2)	= 4.7332	ECPCE (3)	= 388.16	XPCPCRC(1)	= 825.83	XPCPCRC(2)	= -13.895
XPCPCRC(3)	= -38.880	XPCPCDC(1)	= 804.48	XPCPCDC(2)	= -13.127	XPCPCDC(3)	= -36.878
TIMF =	1.400						
UDST (1)	= 79.343	UDST (2)	= 940.55	UDST (3)	= -1451.1	USTSE (1)	= 355.42
USTSE (2)	= -115.84	USTSE (3)	= 1.1896	SRPSE (1)	= 865.01	SRPSE (2)	= -15.542
SRPSE (3)	= -37.373	WDST (1)	= -13.261	WDST (2)	= -1064.0	WDST (3)	= 2114.6
WSTSE (1)	= 280.28	WSTSE (2)	= -319.74	WSTSE (3)	= -137.04	ESTSE (1)	= 11.374
ESTSE (2)	= -10.761	ESTSE (3)	= 400.92	RADAM	= .27400	DREAM	= .18743E-01
CF CT	= 0.	UCPCE (1)	= 361.53	UCPCE (2)	= -111.91	UCPCE (3)	= 7.7247
XCPCE (1)	= 865.78	XCPCE (2)	= -14.671	XCPCE (3)	= -38.023	ECPCE (1)	= 11.525
ECPCE (2)	= -10.462	ECPCE (3)	= 401.04	XPCPCRC(1)	= 864.78	XPCPCRC(2)	= -13.919
XPCPCRC(3)	= -39.257	XPCPCDC(1)	= 843.34	XPCPCDC(2)	= -14.141	XPCPCDC(3)	= -37.635
TIME =	1.500						
UDST (1)	= -1082.0	UDST (2)	= -1129.4	UDST (3)	= -890.34	USTSE (1)	= 300.63
USTSE (2)	= -139.72	USTSE (3)	= -134.97	SRPSE (1)	= 901.51	SRPSE (2)	= -16.914
SRPSE (3)	= -38.044	WDST (1)	= -178.38	WDST (2)	= 733.82	WDST (3)	= 972.82
WSTSE (1)	= 282.82	WSTSE (2)	= -306.98	WSTSE (3)	= 105.87	ESTSE (1)	= -18.784
ESTSE (2)	= -27.680	ESTSE (3)	= 438.17	RADAM	= .22049	DREAM	= -.89814
CF CT	= 0.	UCPCE (1)	= 306.21	UCPCE (2)	= -132.92	UCPCE (3)	= -131.11
XCPCE (1)	= 902.60	XCPCE (2)	= -16.162	XCPCE (3)	= -37.880	ECPCE (1)	= -18.833

Figure 9. (Continued)

```
ECPCE   (2)     = -27.663     ECPCE   (3)     =  438.10     XPCPCRC(1)      =  902.30     XPCPCRC(2)      = -14.606
XPCPCRC(3)      = -38.608     XPCPCDC(1)      =  880.31     XPCPCDC(2)      = -15.019     XPCPCDC(3)      = -37.951

        TIME =  1.600
UDST    (1)     = -373.18     UDST    (2)     = -274.81     UDST    (3)     =  311.97     USTSE   (1)     =  203.93
USTSE   (2)     = -227.57     USTSE   (3)     = -154.98     SRPSE   (1)     =  936.57     SRPSE   (2)     = -18.015
SRPSE   (3)     = -30.313     WDST    (1)     = -444.81     WDST    (2)     =  1424.4     WDST    (3)     = -3593.5
WSTSE   (1)     =  241.38     WSTSE   (2)     = -193.86     WSTSE   (3)     = -53.437     ESTSE   (1)     = -48.121
ESTSE   (2)     = -30.854     ESTSE   (3)     =  480.11     RADAM           =  .41078     DREAM           = -1.3488
CF CT           = 0.          UCPCE   (1)     =  206.65     UCPCE   (2)     = -224.31     UCPCE   (3)     = -152.69
XCPCE   (1)     =  937.52     XCPCE   (2)     = -17.885     XCPCE   (3)     = -37.469     ECPCE   (1)     = -48.322
ECPCE   (2)     = -30.693     ECPCE   (3)     =  480.25     XPCPCRC(1)      =  937.60     XPCPCRC(2)      = -15.835
XPCPCRC(3)      = -37.373     XPCPCDC(1)      =  915.46     XPCPCDC(2)      = -16.129     XPCPCDC(3)      = -37.847

        TIME =  1.700
UDST    (1)     =  844.38     UDST    (2)     =  1249.1     UDST    (3)     =  576.82     USTSE   (1)     =  250.68
USTSE   (2)     = -177.21     USTSE   (3)     = -107.80     SRPSE   (1)     =  989.91     SRPSE   (2)     = -19.090
SRPSE   (3)     = -37.704     WDST    (1)     = -100.55     WDST    (2)     =  1746.0     WDST    (3)     = -1205.9
WSTSE   (1)     =  214.36     WSTSE   (2)     = -29.078     WSTSE   (3)     = -351.12     ESTSE   (1)     = -41.387
ESTSE   (2)     = -7.5414     ESTSE   (3)     =  501.74     RADAM           =  .27916     DREAM           = -1.3257
CF CT           = 0.          UCPCE   (1)     =  251.15     UCPCE   (2)     = -178.01     UCPCE   (3)     = -107.28
XCPCE   (1)     =  970.84     XCPCE   (2)     = -19.010     XCPCE   (3)     = -36.850     ECPCE   (1)     = -41.438
ECPCE   (2)     = -7.5608     ECPCE   (3)     =  501.81     XPCPCRC(1)      =  970.54     XPCPCRC(2)      = -17.594
XPCPCRC(3)      = -35.880     XPCPCDC(1)      =  948.66     XPCPCDC(2)      = -17.405     XPCPCDC(3)      = -37.368

    MORTAR   IGNITION AT TIME =     1.7635 SEC

        TIME =  1.800
UDST    (1)     =  134.78     UDST    (2)     =  1038.9     UDST    (3)     =  866.27     USTSE   (1)     =  308.16
USTSE   (2)     = -41.271     USTSE   (3)     = -37.737     SRPSE   (1)     =  1001.8     SRPSE   (2)     = -20.310
SRPSE   (3)     = -36.978     WDST    (1)     = -936.30     WDST    (2)     =  1119.8     WDST    (3)     =  2444.0
WSTSE   (1)     =  220.09     WSTSE   (2)     =  121.37     WSTSE   (3)     = -224.82     ESTSE   (1)     = -11.303
ESTSE   (2)     =  3.4764     ESTSE   (3)     =  523.94     RADAM           =  .25761     DREAM           = -1.3728
CF CT           = 0.          UCPCE   (1)     =  305.78     UCPCE   (2)     = -39.336     UCPCE   (3)     = -39.987
XCPCE   (1)     =  1002.6     XCPCE   (2)     = -20.188     XCPCE   (3)     = -35.974     ECPCE   (1)     = -11.229
ECPCE   (2)     =  3.5376     ECPCE   (3)     =  524.02     XPCPCRC(1)      =  1001.8     XPCPCRC(2)      = -19.504
XPCPCRC(3)      = -34.095     XPCPCDC(1)      =  980.42     XPCPCDC(2)      = -19.127     XPCPCDC(3)      = -36.335

    MORTAR   STRIPOFF AT TIME =     1.8185 SEC

    RECOVERY CHUTE LAUNCH AT TIME =     1.8185 SEC

        TIME =  1.900
UDST    (1)     = -486.23     UDST    (2)     =  1442.5     UDST    (3)     =  189.10     USTSE   (1)     =  292.10
USTSE   (2)     =  65.883     USTSE   (3)     =  14.701     SRPSE   (1)     =  1032.4     SRPSE   (2)     = -21.488
SRPSE   (3)     = -36.446     WDST    (1)     = -510.06     WDST    (2)     = -318.12     WDST    (3)     = -2626.5
WSTSE   (1)     =  50.154     WSTSE   (2)     =  51.313     WSTSE   (3)     = -279.88     ESTSE   (1)     =  10.379
ESTSE   (2)     = -3.0189     ESTSE   (3)     =  536.04     RADAM           =  .19367     DREAM           =  2.2248
CF CT           = 0.          UCPCE   (1)     =  291.46     UCPCE   (2)     =  62.806     UCPCE   (3)     =  14.880
XCPCE   (1)     =  1033.1     XCPCE   (2)     = -21.318     XCPCE   (3)     = -35.339     ECPCE   (1)     =  10.297
ECPCE   (2)     = -3.1637     ECPCE   (3)     =  536.01     XPCPCRC(1)      =  1031.6     XPCPCRC(2)      = -21.208
XPCPCRC(3)      = -27.408     XPCPCDC(1)      =  1010.9     XPCPCDC(2)      = -20.479     XPCPCDC(3)      = -35.897

    DRAG     CHUTE LINES SEVERED AT TIME =     1.9135 SEC

    SEAT/CREWPERSON SEPARATION AT TIME =     1.9730 SEC

        TIME =  2.000
UDST    (1)     = -2263.0     UDST    (2)     =  1745.4     UDST    (3)     = -103.93     USTSE   (1)     =  193.33
USTSE   (2)     =  212.41     USTSE   (3)     =  13.602     SRPSE   (1)     =  1081.9     SRPSE   (2)     = -22.773
```

Figure 9. (Continued)

```
SRPSE  (3)   = -35.907    WDST   (1)   = -1817.6    WDST   (2)   =  212.95       WDST   (3)   = -9546.3
WSTSE  (1)   =  36.044    WSTSE  (2)   = -5.8713    WSTSE  (3)   = -504.88       ESTSE  (1)   =  45.177
ESTSE  (2)   = -4.4754    ESTSE  (3)   =  539.36    RADAM        =  .96445E-01   DREAM        =  .92122
CF CT        = 0.         UCPCE  (1)   =  209.14    UCPCE  (2)   =  202.13       UCPCE  (3)   =  13.414
XCPCE  (1)   =  1062.5    XCPCE  (2)   = -22.366    XCPCE  (3)   = -34.734       ECPCE  (1)   =  41.933
ECPCE  (2)   = -5.3475    ECPCE  (3)   =  539.75    XPCPCRC(1)   =  1061.3       XPCPCRC(2)   = -22.974
XPCPCRC(3)   = -25.818    XPCPCDC(1)   =  1034.8    XPCPCDC(2)   = -21.165       XPCPCDC(3)   = -35.264

        TIME =  2.100
UDST   (1)   = -2532.4    UDST   (2)   = -2988.5    UDST   (3)   = -1690.6       USTSE  (1)   = -184.61
USTSE  (2)   =  189.55    USTSE  (3)   = -24.592    SRPSE  (1)   =  1089.6       SRPSE  (2)   = -24.023
SRPSE  (3)   = -35.810    WDST   (1)   =  1080.1    WDST   (2)   =  3303.2       WDST   (3)   =  3782.5
WSTSE  (1)   =  87.288    WSTSE  (2)   =  417.07    WSTSE  (3)   = -855.92       ESTSE  (1)   =  133.97
ESTSE  (2)   = -14.950    ESTSE  (3)   =  530.82    RADAM        =  .11112       DREAM        =  .91967
CF CT        = 0.         UCPCE  (1)   =  88.832    UCPCE  (2)   =  277.03       UCPCE  (3)   =  4.4810
XCPCE  (1)   =  1091.3    XCPCE  (2)   = -23.390    XCPCE  (3)   = -33.827       ECPCE  (1)   =  74.245
ECPCE  (2)   = -9.6593    ECPCE  (3)   =  539.16    XPCPCRC(1)   =  1088.5       XPCPCRC(2)   = -24.133
XPCPCRC(3)   = -29.758    XPCPCDC(1)   =  1048.5    XPCPCDC(2)   = -21.513       XPCPCDC(3)   = -34.804

        TIME =  2.200
UDST   (1)   =  2220.5    UDST   (2)   = -1458.7    UDST   (3)   = -982.55       USTSE  (1)   = -140.19
USTSE  (2)   = -73.848    USTSE  (3)   = -205.15    SRPSE  (1)   =  1115.8       SRPSE  (2)   = -25.519
SRPSE  (3)   = -34.802    WDST   (1)   =  1149.4    WDST   (2)   = -1097.3       WDST   (3)   =  4544.5
WSTSE  (1)   =  105.59    WSTSE  (2)   =  449.87    WSTSE  (3)   = -447.45       ESTSE  (1)   =  222.50
ESTSE  (2)   = -35.003    ESTSE  (3)   =  500.41    RADAM        =  .11934       DREAM        =  1.3545
CF CT        = 0.         UCPCE  (1)   = -86.364    UCPCE  (2)   =  268.42       UCPCE  (3)   =  13.115
XCPCE  (1)   =  1118.5    XCPCE  (2)   = -24.335    XCPCE  (3)   = -32.528       ECPCE  (1)   =  107.30
ECPCE  (2)   = -12.704    ECPCE  (3)   =  529.73    XPCPCRC(1)   =  1112.7       XPCPCRC(2)   = -24.840
XPCPCRC(3)   = -34.005    XPCPCDC(1)   =  1054.3    XPCPCDC(2)   = -21.741       XPCPCDC(3)   = -34.234

        TIME =  2.300
UDST   (1)   =  982.20    UDST   (2)   = -1041.0    UDST   (3)   =  1290.4       USTSE  (1)   =  49.146
USTSE  (2)   = -166.23    USTSE  (3)   = -186.48    SRPSE  (1)   =  1141.2       SRPSE  (2)   = -27.145
SRPSE  (3)   = -33.584    WDST   (1)   =  4362.7    WDST   (2)   = -1884.8       WDST   (3)   =  5932.5
WSTSE  (1)   =  341.83    WSTSE  (2)   =  330.01    WSTSE  (3)   =  22.632       ESTSE  (1)   =  279.53
ESTSE  (2)   = -43.281    ESTSE  (3)   =  485.13    RADAM        =  .13430       DREAM        =  1.8022
CF CT        = 0.         UCPCE  (1)   = -218.89    UCPCE  (2)   =  165.02       UCPCE  (3)   =  30.337
XCPCE  (1)   =  1147.2    XCPCE  (2)   = -25.208    XCPCE  (3)   = -30.875       ECPCE  (1)   =  141.89
ECPCE  (2)   = -6.6246    ECPCE  (3)   =  514.54    XPCPCRC(1)   =  1134.6       XPCPCRC(2)   = -25.291
XPCPCRC(3)   = -34.753    XPCPCDC(1)   =  1080.2    XPCPCDC(2)   = -21.909       XPCPCDC(3)   = -33.527

        TIME =  2.400
UDST   (1)   = -2229.5    UDST   (2)   =  1357.3    UDST   (3)   =  2824.7       USTSE  (1)   = -53.026
USTSE  (2)   = -232.20    USTSE  (3)   =  55.009    SRPSE  (1)   =  1168.2       SRPSE  (2)   = -28.186
SRPSE  (3)   = -32.161    WDST   (1)   = -1373.3    WDST   (2)   = -3342.8       WDST   (3)   =  1643.1
WSTSE  (1)   =  729.93    WSTSE  (2)   = -21.214    WSTSE  (3)   =  587.26       ESTSE  (1)   =  233.57
ESTSE  (2)   = -60.511    ESTSE  (3)   =  586.25    RADAM        =  .15016       DREAM        =  1.7225
CF CT        = 0.         UCPCE  (1)   = -265.52    UCPCE  (2)   = -.10516       UCPCE  (3)   =  35.260
XCPCE  (1)   =  1174.2    XCPCE  (2)   = -26.058    XCPCE  (3)   = -28.807       ECPCE  (1)   =  173.49
ECPCE  (2)   =  10.483    ECPCE  (3)   =  501.06    XPCPCRC(1)   =  1155.6       XPCPCRC(2)   = -26.191
XPCPCRC(3)   = -33.247    XPCPCDC(1)   =  1064.8    XPCPCDC(2)   = -22.044       XPCPCDC(3)   = -32.676

    RECOVERY CHUTE LINESTRETCH AT TIME =     2.4640 SEC


        TIME =  2.500
UDST   (1)   =  1283.8    UDST   (2)   =  3045.8    UDST   (3)   = -461.84       USTSE  (1)   = -113.99
USTSE  (2)   =  63.836    USTSE  (3)   =  184.22    SRPSE  (1)   =  1189.6       SRPSE  (2)   = -28.767
SRPSE  (3)   = -30.055    WDST   (1)   = -1018.3    WDST   (2)   =  380.95       WDST   (3)   = -744.28
WSTSE  (1)   =  582.20    WSTSE  (2)   = -136.24    WSTSE  (3)   =  650.81       ESTSE  (1)   =  237.91
ESTSE  (2)   =  .11557    ESTSE  (3)   =  655.78    RADAM        =  .19229       DREAM        =  2.0931
CF CT        = 0.         UCPCE  (1)   = -208.89    UCPCE  (2)   = -158.90       UCPCE  (3)   =  9.4201
XCPCE  (1)   =  1200.6    XCPCE  (2)   = -26.848    XCPCE  (3)   = -26.609       ECPCE  (1)   =  202.16
ECPCE  (2)   =  35.925    ECPCE  (3)   =  494.72    XPCPCRC(1)   =  1179.8       XPCPCRC(2)   = -27.540
XPCPCRC(3)   = -29.676    XPCPCDC(1)   =  1068.6    XPCPCDC(2)   = -22.155       XPCPCDC(3)   = -31.676
```

Figure 9. (Continued)

```
         TIME =  2.600
UDST     (1)    =  1309.9      UDST     (2)    = -261.83      UDST     (3)    = -1847.6      USTSE    (1)    =  68.214
USTSE    (2)    =  190.84      USTSE    (3)    =  36.117      SRPSE    (1)    =  1211.4      SRPSE    (2)    = -29.563
SRPSE    (3)    = -27.780      WDST     (1)    = -1710.8      WDST     (2)    = -2322.3      WDST     (3)    = -4511.2
WSTSE    (1)    =  401.06      WSTSE    (2)    = -150.73      WSTSE    (3)    =  396.45      ESTSE    (1)    =  289.78
ESTSE    (2)    =  10.754      ESTSE    (3)    =  723.07      RADAM           =  .29896      DREAM           =  4.3204
CF CT           = 0.           UCPCE    (1)    = -11.070      UCPCE    (2)    = -175.18      UCPCE    (3)    =  163.52
XCPCE    (1)    =  1225.5      XCPCE    (2)    = -27.565      XCPCE    (3)    = -24.120      ECPCE    (1)    =  282.10
ECPCE    (2)    =  70.543      ECPCE    (3)    =  597.75      XPCPCRC(1)      =  1201.0      XPCPCRC(2)      = -28.037
XPCPCRC(3)      = -25.902      XPCPCDC(1)      =  1071.9      XPCPCDC(2)      = -22.250      XPCPCDC(3)      = -30.526

    RECOVERY CHUTE REEFED AT TIME =     2.6265 SEC

         TIME =  2.700
UDST     (1)    = -1054.5      UDST     (2)    =  4.0285      UDST     (3)    = -674.72      USTSE    (1)    =  79.103
USTSE    (2)    =  150.90      USTSE    (3)    = -100.37      SRPSE    (1)    =  1232.0      SRPSE    (2)    = -30.196
SRPSE    (3)    = -25.402      WDST     (1)    = -3707.3      WDST     (2)    =  733.53      WDST     (3)    = -6706.7
WSTSE    (1)    =  134.11      WSTSE    (2)    = -201.65      WSTSE    (3)    = -195.47      ESTSE    (1)    =  201.20
ESTSE    (2)    = -4.5534      ESTSE    (3)    =  750.01      RADAM           =  .77105      DREAM           =  13.760
CF CT           = 0.           UCPCE    (1)    = -86.304      UCPCE    (2)    =  120.91      UCPCE    (3)    =  144.61
XCPCE    (1)    =  1247.7      XCPCE    (2)    = -28.238      XCPCE    (3)    = -21.633      ECPCE    (1)    =  241.25
ECPCE    (2)    =  40.201      ECPCE    (3)    =  657.65      XPCPCRC(1)      =  1223.9      XPCPCRC(2)      = -28.396
XPCPCRC(3)      = -23.895      XPCPCDC(1)      =  1074.7      XPCPCDC(2)      = -22.332      XPCPCDC(3)      = -29.229

         TIME =  2.800
UDST     (1)    = -1834.7      UDST     (2)    = -996.53      UDST     (3)    =  370.80      USTSE    (1)    = -90.879
USTSE    (2)    =  145.24      USTSE    (3)    = -98.068      SRPSE    (1)    =  1251.9      SRPSE    (2)    = -30.602
SRPSE    (3)    = -22.772      WDST     (1)    = -1026.8      WDST     (2)    =  2496.1      WDST     (3)    = -2721.8
WSTSE    (1)    = -100.34      WSTSE    (2)    = -51.322      WSTSE    (3)    = -712.81      ESTSE    (1)    =  242.26
ESTSE    (2)    =  14.139      ESTSE    (3)    =  755.92      RADAM           =  .71071      DREAM           =  12.611
CF CT           = 0.           UCPCE    (1)    = -111.40      UCPCE    (2)    = -7.0328      UCPCE    (3)    =  148.49
XCPCE    (1)    =  1267.2      XCPCE    (2)    = -29.138      XCPCE    (3)    = -18.075      ECPCE    (1)    =  230.40
ECPCE    (2)    =  4.3998      ECPCE    (3)    =  633.95      XPCPCRC(1)      =  1243.3      XPCPCRC(2)      = -27.171
XPCPCRC(3)      = -22.307      XPCPCDC(1)      =  1077.2      XPCPCDC(2)      = -22.404      XPCPCDC(3)      = -27.780

         TIME =  2.900
UDST     (1)    =  785.48      UDST     (2)    = -1835.7      UDST     (3)    = -495.66      USTSE    (1)    = -157.13
USTSE    (2)    = -34.909      USTSE    (3)    = -102.00      SRPSE    (1)    =  1271.1      SRPSE    (2)    = -31.125
SRPSE    (3)    = -20.106      WDST     (1)    =  470.34      WDST     (2)    =  1567.8      WDST     (3)    =  1965.6
WSTSE    (1)    = -103.45      WSTSE    (2)    =  169.06      WSTSE    (3)    = -728.66      ESTSE    (1)    =  150.92
ESTSE    (2)    =  37.445      ESTSE    (3)    =  702.73      RADAM           =  .28494      DREAM           =  4.7739
CF CT           = 0.           UCPCE    (1)    =  80.033      UCPCE    (2)    = -5.9804      UCPCE    (3)    =  146.14
XCPCE    (1)    =  1284.6      XCPCE    (2)    = -30.077      XCPCE    (3)    = -16.519      ECPCE    (1)    =  295.09
ECPCE    (2)    = -29.630      ECPCE    (3)    =  621.20      XPCPCRC(1)      =  1260.7      XPCPCRC(2)      = -27.889
XPCPCRC(3)      = -19.686      XPCPCDC(1)      =  1079.4      XPCPCDC(2)      = -22.468      XPCPCDC(3)      = -26.209

    RECOVERY CHUTE DISREEFED AT TIME =     2.9135 SEC

         TIME =  3.000
UDST     (1)    =  1564.6      UDST     (2)    = -54.833      UDST     (3)    = -44.189      USTSE    (1)    = -12.106
USTSE    (2)    = -122.47      USTSE    (3)    = -140.06      SRPSE    (1)    =  1289.8      SRPSE    (2)    = -31.450
SRPSE    (3)    = -17.412      WDST     (1)    =  1093.8      WDST     (2)    =  657.01      WDST     (3)    =  3426.4
WSTSE    (1)    = -15.230      WSTSE    (2)    =  263.75      WSTSE    (3)    = -434.75      ESTSE    (1)    =  90.657
ESTSE    (2)    =  15.789      ESTSE    (3)    =  663.16      RADAM           =  .20463      DREAM           =  3.3726
CF CT           = 0.           UCPCE    (1)    =  40.128      UCPCE    (2)    =  74.331      UCPCE    (3)    =  124.48
XCPCE    (1)    =  1300.3      XCPCE    (2)    = -30.859      XCPCE    (3)    = -13.969      ECPCE    (1)    =  282.68
ECPCE    (2)    = -64.959      ECPCE    (3)    =  651.08      XPCPCRC(1)      =  1276.4      XPCPCRC(2)      = -29.927
XPCPCRC(3)      = -17.183      XPCPCDC(1)      =  1081.3      XPCPCDC(2)      = -22.525      XPCPCDC(3)      = -24.499

         TIME =  3.100
UDST     (1)    =  619.33      UDST     (2)    = -10.808      UDST     (3)    =  741.20      USTSE    (1)    =  100.32
USTSE    (2)    = -113.54      USTSE    (3)    = -101.36      SRPSE    (1)    =  1308.1      SRPSE    (2)    = -31.806
```

Figure 9. (Continued)

```
SRPSE   (3)    = -14.488    WDST    (1)   =  1196.0    WDST    (2)   = -419.78    WDST    (3)   =  2859.7
WSTSE   (1)    =  110.89    WSTSE   (2)   =  273.67    WSTSE   (3)   = -101.40    ESTSE   (1)   =  52.098
ESTSE   (2)    =  7.2831    ESTSE   (3)   =  881.59    RADAM         =  .81849    DREAM         =  14.899
CF CT          = 0.         UCPCE   (1)   = -20.155    UCPCE   (2)   = -33.793    UCPCE   (3)   =  124.52
XCPCE   (1)    =  1314.1    XCPCE   (2)   = -31.525    XCPCE   (3)   = -11.309    ECPCE   (1)   =  104.34
ECPCE   (2)    = -68.897    ECPCE   (3)   =  687.52    XPCPCRC(1)    =  1289.8    XPCPCRC(2)    = -30.812
XPCPCRC(3)     = -15.340    XPCPCDC(1)    =  1083.1    XPCPCDC(2)    = -22.578    XPCPCDC(3)    = -22.884

          TIME =  3.200
UDST    (1)    = -423.48    UDST    (2)   = -348.93    UDST    (3)   =  929.46    USTSE   (1)   =  110.60
USTSE   (2)    = -134.88    USTSE   (3)   = -14.181    SRPSE   (1)   =  1325.8    SRPSE   (2)   = -32.377
SRPSE   (3)    = -11.320    WDST    (1)   =  801.27    WDST    (2)   = -977.09    WDST    (3)   =  2615.8
WSTSE   (1)    =  220.68    WSTSE   (2)   =  195.29    WSTSE   (3)   =  171.55    ESTSE   (1)   =  35.739
ESTSE   (2)    =  24.114    ESTSE   (3)   =  674.93    RADAM         =  .86087E-01 DREAM        =  .36811
CF CT          = 0.         UCPCE   (1)   =  38.995    UCPCE   (2)   = -41.587    UCPCE   (3)   =  94.728
XCPCE   (1)    =  1325.8    XCPCE   (2)   = -32.247    XCPCE   (3)   = -8.5830    ECPCE   (1)   =  68.177
ECPCE   (2)    = -64.330    ECPCE   (3)   =  798.07    XPCPCRC(1)    =  1301.9    XPCPCRC(2)    = -30.770
XPCPCRC(3)     = -13.308    XPCPCDC(1)    =  1084.6    XPCPCDC(2)    = -22.621    XPCPCDC(3)    = -20.716

          TIME =  3.300
UDST    (1)    = -1081.0    UDST    (2)   =  198.48    UDST    (3)   =  698.11    USTSE   (1)   =  25.893
USTSE   (2)    = -150.33    USTSE   (3)   =  70.120    SRPSE   (1)   =  1342.4    SRPSE   (2)   = -32.958
SRPSE   (3)    = -7.8398    WDST    (1)   =  44.921    WDST    (2)   =  671.70    WDST    (3)   =  652.62
WSTSE   (1)    =  246.52    WSTSE   (2)   =  121.08    WSTSE   (3)   =  346.94    ESTSE   (1)   =  58.066
ESTSE   (2)    =  48.147    ESTSE   (3)   =  714.77    RADAM         =  .40504    DREAM         =  7.0091
CF CT          = 0.         UCPCE   (1)   = -8.0281    UCPCE   (2)   =  25.893    UCPCE   (3)   =  89.942
XCPCE   (1)    =  1335.6    XCPCE   (2)   = -32.872    XCPCE   (3)   = -5.8125    ECPCE   (1)   =  98.848
ECPCE   (2)    = -15.168    ECPCE   (3)   =  806.62    XPCPCRC(1)    =  1311.6    XPCPCRC(2)    = -31.058
XPCPCRC(3)     = -11.147    XPCPCDC(1)    =  1086.0    XPCPCDC(2)    = -22.862    XPCPCDC(3)    = -18.864

          TIME =  3.400
UDST    (1)    = -891.45    UDST    (2)   =  981.65    UDST    (3)   =  24.681    USTSE   (1)   = -79.477
USTSE   (2)    = -88.025    USTSE   (3)   =  111.52    SRPSE   (1)   =  1358.5    SRPSE   (2)   = -33.431
SRPSE   (3)    = -4.1533    WDST    (1)   =  4.0778    WDST    (2)   =  893.56    WDST    (3)   = -655.05
WSTSE   (1)    =  231.69    WSTSE   (2)   =  211.81    WSTSE   (3)   =  338.40    ESTSE   (1)   =  118.99
ESTSE   (2)    =  45.145    ESTSE   (3)   =  783.02    RADAM         =  .34037    DREAM         =  5.9576
CF CT          = 0.         UCPCE   (1)   = -38.673    UCPCE   (2)   = -13.861    UCPCE   (3)   =  87.486
XCPCE   (1)    =  1343.7    XCPCE   (2)   = -33.490    XCPCE   (3)   = -3.0871    ECPCE   (1)   =  107.81
ECPCE   (2)    =  32.782    ECPCE   (3)   =  790.93    XPCPCRC(1)    =  1320.2    XPCPCRC(2)    = -32.044
XPCPCRC(3)     = -8.9890    XPCPCDC(1)    =  1087.3    XPCPCDC(2)    = -22.899    XPCPCDC(3)    = -16.518

          TIME =  3.500
UDST    (1)    = -250.19    UDST    (2)   =  814.20    UDST    (3)   = -795.19    USTSE   (1)   = -138.38
USTSE   (2)    =  14.820    USTSE   (3)   =  70.620    SRPSE   (1)   =  1373.9    SRPSE   (2)   = -33.799
SRPSE   (3)    = -.34002    WDST    (1)   =  739.70    WDST    (2)   =  258.92    WDST    (3)   = -802.77
WSTSE   (1)    =  254.64    WSTSE   (2)   =  280.70    WSTSE   (3)   =  238.09    ESTSE   (1)   =  150.02
ESTSE   (2)    =  18.186    ESTSE   (3)   =  825.25    RADAM         =  .12328    DREAM         =  1.7460
CF CT          = 0.         UCPCE   (1)   =  3.0473    UCPCE   (2)   = -10.043    UCPCE   (3)   =  65.958
XCPCE   (1)    =  1350.3    XCPCE   (2)   = -33.770    XCPCE   (3)   = -.32694    ECPCE   (1)   =  33.183
ECPCE   (2)    =  58.262    ECPCE   (3)   =  742.89    XPCPCRC(1)    =  1327.0    XPCPCRC(2)    = -32.766
XPCPCRC(3)     = -7.2440    XPCPCDC(1)    =  1088.4    XPCPCDC(2)    = -22.731    XPCPCDC(3)    = -14.288

          TIME =  3.600
UDST    (1)    =  358.93    UDST    (2)   =  229.41    UDST    (3)   = -893.78    USTSE   (1)   = -129.84
USTSE   (2)    =  75.619    USTSE   (3)   = -28.220    SRPSE   (1)   =  1388.8    SRPSE   (2)   = -34.291
SRPSE   (3)    =  3.7174    WDST    (1)   =  739.53    WDST    (2)   = -1554.0    WDST    (3)   = -329.75
WSTSE   (1)    =  342.73    WSTSE   (2)   =  219.94    WSTSE   (3)   =  173.82    ESTSE   (1)   =  163.03
ESTSE   (2)    = -12.178    ESTSE   (3)   =  856.56    RADAM         =  .16721    DREAM         =  2.9083
CF CT          = 0.         UCPCE   (1)   = -15.815    UCPCE   (2)   =  14.982    UCPCE   (3)   =  52.388
XCPCE   (1)    =  1355.8    XCPCE   (2)   = -33.909    XCPCE   (3)   =  2.3726    ECPCE   (1)   = -78.357
ECPCE   (2)    =  56.350    ECPCE   (3)   =  672.13    XPCPCRC(1)    =  1332.9    XPCPCRC(2)    = -32.993
XPCPCRC(3)     = -5.4790    XPCPCDC(1)    =  1089.4    XPCPCDC(2)    = -22.761    XPCPCDC(3)    = -11.985

          TIME =  3.700
UDST    (1)    =  203.15    UDST    (2)   = -413.55    UDST    (3)   = -440.10    USTSE   (1)   = -94.342
```

Figure 9. (Continued)

```
USTSE   (2)   =  61.517    USTSE   (3)   = -101.49    SRPSE   (1)   =  1403.3    SRPSE   (2)   = -35.027
SRPSE   (3)   =  8.0999    WDST    (1)   = -917.36    WDST    (2)   = -1756.2    WDST    (3)   = -918.77
WSTSE   (1)   =  329.88    WSTSE   (2)   =  35.048    WSTSE   (3)   =  113.75    ESTSE   (1)   =  155.33
ESTSE   (2)   = -29.709    ESTSE   (3)   =  895.10    RADAM         =  .24063    DREAM         =  4.1740
CF CT         = 0.         UCPCE   (1)   = -21.651    UCPCE   (2)   = -15.382    UCPCE   (3)   =  40.788
XCPCE   (1)   =  1360.5    XCPCE   (2)   = -34.032    XCPCE   (3)   =  4.9679    ECPCE   (1)   = -115.41
ECPCE   (2)   =  12.578    ECPCE   (3)   =  638.64    XPCPCRC(1)    =  1338.0    XPCPCRC(2)    = -32.954
XPCPCRC(3)    = -3.6329    XPCPCDC(1)    =  1090.3    XPCPCDC(2)    = -22.787    XPCPCDC(3)    = -9.6180

        TIME =  3.800
UDST    (1)   = -189.41    UDST    (2)   = -431.14    UDST    (3)   =  91.027    USTSE   (1)   = -95.880
USTSE   (2)   =  15.790    USTSE   (3)   = -116.88    SRPSE   (1)   =  1417.7    SRPSE   (2)   = -35.925
SRPSE   (3)   =  12.806    WDST    (1)   = -922.17    WDST    (2)   = -1149.6    WDST    (3)   = -643.35
WSTSE   (1)   =  227.40    WSTSE   (2)   = -101.73    WSTSE   (3)   =  20.414    ESTSE   (1)   =  149.22
ESTSE   (2)   = -25.076    ESTSE   (3)   =  926.00    RADAM         =  .11467    DREAM         =  1.2085
CF CT         = 0.         UCPCE   (1)   =  13.088    UCPCE   (2)   = -19.852    UCPCE   (3)   =  35.727
XCPCE   (1)   =  1364.2    XCPCE   (2)   = -34.205    XCPCE   (3)   =  7.5082    ECPCE   (1)   = -102.50
ECPCE   (2)   = -42.311    ECPCE   (3)   =  834.30    XPCPCRC(1)    =  1342.0    XPCPCRC(2)    = -32.980
XPCPCRC(3)    = -1.8607    XPCPCDC(1)    =  1091.2    XPCPCDC(2)    = -22.811    XPCPCDC(3)    = -7.1952

        TIME =  3.900
UDST    (1)   = -301.55    UDST    (2)   = -277.89    UDST    (3)   =  473.90    USTSE   (1)   = -122.69
USTSE   (2)   = -20.163    USTSE   (3)   = -88.151    SRPSE   (1)   =  1432.0    SRPSE   (2)   = -36.882
SRPSE   (3)   =  17.765    WDST    (1)   = -461.34    WDST    (2)   = -982.50    WDST    (3)   = -81.925
WSTSE   (1)   =  181.45    WSTSE   (2)   = -204.90    WSTSE   (3)   = -6.2180    ESTSE   (1)   =  158.13
ESTSE   (2)   = -12.215    ESTSE   (3)   =  942.35    RADAM         =  .10780    DREAM         =  .22738
CF CT         = 0.         UCPCE   (1)   =  19.129    UCPCE   (2)   =  11.496    UCPCE   (3)   =  31.737
XCPCE   (1)   =  1367.4    XCPCE   (2)   = -34.438    XCPCE   (3)   =  10.084    ECPCE   (1)   = -217.14
ECPCE   (2)   = -75.129    ECPCE   (3)   =  766.72    XPCPCRC(1)    =  1345.5    XPCPCRC(2)    = -33.172
XPCPCRC(3)    = -.10644    XPCPCDC(1)    =  1091.9    XPCPCDC(2)    = -22.832    XPCPCDC(3)    = -4.7248

        TIME =  4.000
UDST    (1)   = -70.998    UDST    (2)   = -183.77    UDST    (3)   =  737.23    USTSE   (1)   = -144.70
USTSE   (2)   = -40.514    USTSE   (3)   = -25.014    SRPSE   (1)   =  1448.3    SRPSE   (2)   = -37.801
SRPSE   (3)   =  23.004    WDST    (1)   = -1009.2    WDST    (2)   = -302.83    WDST    (3)   = -672.63
WSTSE   (1)   =  82.944    WSTSE   (2)   = -277.87    WSTSE   (3)   = -45.001    ESTSE   (1)   =  178.08
ESTSE   (2)   =  2.4924    ESTSE   (3)   =  953.64    RADAM         =  .21718    DREAM         =  3.2529
CF CT         = 0.         UCPCE   (1)   = -4.0228    UCPCE   (2)   =  17.804    UCPCE   (3)   =  30.829
XCPCE   (1)   =  1370.0    XCPCE   (2)   = -34.674    XCPCE   (3)   =  12.772    ECPCE   (1)   = -238.87
ECPCE   (2)   = -17.424    ECPCE   (3)   =  781.29    XPCPCRC(1)    =  1348.7    XPCPCRC(2)    = -33.474
XPCPCRC(3)    =  1.5705    XPCPCDC(1)    =  1092.5    XPCPCDC(2)    = -22.851    XPCPCDC(3)    = -2 2138

        TIME =  4.100
UDST    (1)   =  355.58    UDST    (2)   = -297.07    UDST    (3)   =  528.97    USTSE   (1)   = -129.82
USTSE   (2)   = -62.047    USTSE   (3)   =  42.182    SRPSE   (1)   =  1460.4    SRPSE   (2)   = -38.602
SRPSE   (3)   =  28.483    WDST    (1)   = -804.45    WDST    (2)   =  599.84    WDST    (3)   = -812.87
WSTSE   (1)   = -13.785    WSTSE   (2)   = -254.36    WSTSE   (3)   = -126.96    ESTSE   (1)   =  205.84
ESTSE   (2)   =  10.024    ESTSE   (3)   =  980.44    RADAM         =  .18868    DREAM         =  2.5614
CF CT         = 0.         UCPCE   (1)   = -10.402    UCPCE   (2)   =  1.4492    UCPCE   (3)   =  32.025
XCPCE   (1)   =  1372.0    XCPCE   (2)   = -34.837    XCPCE   (3)   =  15.589    ECPCE   (1)   = -278.81
ECPCE   (2)   =  23.204    ECPCE   (3)   =  758.19    XPCPCRC(1)    =  1351.5    XPCPCRC(2)    = -33.767
XPCPCRC(3)    =  [illegible].1418   XPCPCDC(1)    =  1093.1    XPCPCDC(2)    = -22.868    XPCPCDC(3)    =  .33166

        TIME =  4.200
UDST    (1)   =  575.21    UDST    (2)   = -357.08    UDST    (3)   =  86.841    USTSE   (1)   = -81.096
USTSE   (2)   = -96.789    USTSE   (3)   =  73.184    SRPSE   (1)   =  1474.1    SRPSE   (2)   = -39.302
SRPSE   (3)   =  34.022    WDST    (1)   = -545.95    WDST    (2)   =  794.11    WDST    (3)   = -362.40
WSTSE   (1)   = -79.472    WSTSE   (2)   = -179.64    WSTSE   (3)   = -189.79    ESTSE   (1)   =  233.18
ESTSE   (2)   =  6.5115    ESTSE   (3)   =  959.85    RADAM         =  .10675    DREAM         = -.85751
CF CT         = 0.         UCPCE   (1)   = -2.1191    UCPCE   (2)   = -.93769    UCPCE   (3)   =  32.157
XCPCE   (1)   =  1373.5    XCPCE   (2)   = -34.886    XCPCE   (3)   =  18.483    ECPCE   (1)   = -344.57
ECPCE   (2)   =  28.171    ECPCE   (3)   =  724.69    XPCPCRC(1)    =  1354.0    XPCPCRC(2)    = -33.975
XPCPCRC(3)    =  4.6734    XPCPCDC(1)    =  1093.6    XPCPCDC(2)    = -22.883    XPCPCDC(3)    =  2.9061

        TIME =  4.300
```

Figure 9. (Continued)

```
UDST    (1)    =  564.06     UDST    (2)    = -213.13     UDST    (3)    = -284.92        USTSE   (1)    = -22.668
USTSE   (2)    = -126.12     USTSE   (3)    =  82.203     SRPSE   (1)    =  1487.4        SRPSE   (2)    = -39.995
SRPSE   (3)    =  39.626     WDST    (1)    = -570.74     WDST    (2)    =  806.93        WDST    (3)    =  84.503
WSTSE   (1)    = -132.05     WSTSE   (2)    = -98.660     WSTSE   (3)    = -201.21        ESTSE   (1)    =  255.90
ESTSE   (2)    = -2.1670     ESTSE   (3)    =  950.33     RADAM          =  .72086E-01    DREAM          = -.21392
CF CT          = 0.          UCPCE   (1)    = -7.2367     UCPCE   (2)    = -.19007        UCPCE   (3)    =  30.707
XCPCE   (1)    =  1374.8     XCPCE   (2)    = -34.843     XCPCE   (3)    =  21.421        ECPCE   (1)    = -412.95
ECPCE   (2)    =  25.660     ECPCE   (3)    =  702.27     XPCPCRC(1)     =  1356.2        XPCPCRC(2)     = -34.080
XPCPCRC(3)     =  6.2036     XPCPCDC(1)     =  1094.1     XPCPCDC(2)     = -22.896        XPCPCDC(3)     =  5.5049

        TIME =  4.400
UDST    (1)    =  372.87     UDST    (2)    =  28.749     UDST    (3)    = -480.67        USTSE   (1)    =  28.401
USTSE   (2)    = -135.53     USTSE   (3)    =  22.619     SRPSE   (1)    =  1500.2        SRPSE   (2)    = -40.737
SRPSE   (3)    =  45.354     WDST    (1)    = -483.87     WDST    (2)    = -179.23        WDST    (3)    =  778.51
WSTSE   (1)    = -177.12     WSTSE   (2)    = -38.681     WSTSE   (3)    = -162.26        ESTSE   (1)    =  274.08
ESTSE   (2)    = -10.504     ESTSE   (3)    =  932.83     RADAM          =  .15770        DREAM          =  2.4319
CF CT          = 0.          UCPCE   (1)    = -4.0418     UCPCE   (2)    = -15.552        UCPCE   (3)    =  25.912
XCPCE   (1)    =  1375.7     XCPCE   (2)    = -34.799     XCPCE   (3)    =  24.364        ECPCE   (1)    = -470.91
ECPCE   (2)    =  1.8543     ECPCE   (3)    =  874.38     XPCPCRC(1)     =  1358.3        XPCPCRC(2)     = -34.112
XPCPCRC(3)     =  7.8861     XPCPCDC(1)     =  1094.5     XPCPCDC(2)     = -22.908        XPCPCDC(3)     =  8.1240

        TIME =  4.500
UDST    (1)    = -41.050     UDST    (2)    =  104.97     UDST    (3)    = -450.59        USTSE   (1)    =  44.261
USTSE   (2)    = -126.90     USTSE   (3)    = -25.066     SRPSE   (1)    =  1512.8        SRPSE   (2)    = -41.290
SRPSE   (3)    =  51.126     WDST    (1)    =  586.88     WDST    (2)    = -353.73        WDST    (3)    =  1629.6
WSTSE   (1)    = -164.71     WSTSE   (2)    = -95.003     WSTSE   (3)    = -15.018        ESTSE   (1)    =  285.27
ESTSE   (2)    = -8.0436     ESTSE   (3)    =  912.51     RADAM          =  .14194        DREAM          =  1.5671
CF CT          = 0.          UCPCE   (1)    =  19.591     UCPCE   (2)    = -12.942        UCPCE   (3)    =  17.625
XCPCE   (1)    =  1376.4     XCPCE   (2)    = -34.825     XCPCE   (3)    =  27.281        ECPCE   (1)    = -509.41
ECPCE   (2)    = -49.459     ECPCE   (3)    =  675.84     XPCPCRC(1)     =  1360.1        XPCPCRC(2)     = -34.104
XPCPCRC(3)     =  9.7726     XPCPCDC(1)     =  1094.9     XPCPCDC(2)     = -22.919        XPCPCDC(3)     =  10.760

    RECOVERY CANOPY FILLED AT TIME =     4.5060 SEC

        TIME =  4.600
UDST    (1)    = -460.87     UDST    (2)    =  55.646     UDST    (3)    = -213.94        USTSE   (1)    =  18.287
USTSE   (2)    = -119.12     USTSE   (3)    = -59.469     SRPSE   (1)    =  1525.0        SRPSE   (2)    = -41.572
SRPSE   (3)    =  56.915     WDST    (1)    =  836.73     WDST    (2)    =  261.78        WDST    (3)    =  1666.1
WSTSE   (1)    = -83.767     WSTSE   (2)    = -100.81     WSTSE   (3)    =  161.58        ESTSE   (1)    =  279.11
ESTSE   (2)    =  2.5051     ESTSE   (3)    =  899.03     RADAM          =  .13113        DREAM          = -.43436
CF CT          = 0.          UCPCE   (1)    =  21.550     UCPCE   (2)    =  15.530        UCPCE   (3)    =  12.501
XCPCE   (1)    =  1376.7     XCPCE   (2)    = -34.920     XCPCE   (3)    =  30.183        ECPCE   (1)    = -618.51
ECPCE   (2)    = -49.905     ECPCE   (3)    =  774.56     XPCPCRC(1)     =  1361.7        XPCPCRC(2)     = -34.098
XPCPCRC(3)     =  11.842     XPCPCDC(1)     =  1095.2     XPCPCDC(2)     = -22.928        XPCPCDC(3)     =  13.410

        TIME =  4.700
UDST    (1)    = -604.77     UDST    (2)    =  234.72     UDST    (3)    = -11.884        USTSE   (1)    = -37.790
USTSE   (2)    = -106.92     USTSE   (3)    = -59.600     SRPSE   (1)    =  1537.0        SRPSE   (2)    = -41.690
SRPSE   (3)    =  62.792     WDST    (1)    =  570.62     WDST    (2)    =  490.30        WDST    (3)    =  953.48
WSTSE   (1)    = -18.798     WSTSE   (2)    = -58.052     WSTSE   (3)    =  287.79        ESTSE   (1)    =  255.48
ESTSE   (2)    =  6.8630     ESTSE   (3)    =  892.12     RADAM          =  .15203        DREAM          =  1.5526
CF CT          = 0.          UCPCE   (1)    = -6.0583     UCPCE   (2)    =  26.257        UCPCE   (3)    =  14.727
XCPCE   (1)    =  1378.7     XCPCE   (2)    = -35.008     XCPCE   (3)    =  33.185        ECPCE   (1)    = -645.37
ECPCE   (2)    =  10.137     ECPCE   (3)    =  779.05     XPCPCRC(1)     =  1363.1        XPCPCRC(2)     = -34.120
XPCPCRC(3)     =  13.807     XPCPCDC(1)     =  1095.5     XPCPCDC(2)     = -22.937        XPCPCDC(3)     =  16.071

        TIME =  4.800
UDST    (1)    = -428.86     UDST    (2)    =  576.66     UDST    (3)    =  49.542        USTSE   (1)    = -92.793
USTSE   (2)    = -66.308     USTSE   (3)    = -65.927     SRPSE   (1)    =  1549.8        SRPSE   (2)    = -41.786
SRPSE   (3)    =  68.829     WDST    (1)    =  201.33     WDST    (2)    =  282.26        WDST    (3)    =  358.09
WSTSE   (1)    =  29.859     WSTSE   (2)    = -20.163     WSTSE   (3)    =  365.94        ESTSE   (1)    =  221.72
ESTSE   (2)    =  6.5843     ESTSE   (3)    =  888.22     RADAM          =  .17314        DREAM          =  2.7532
CF CT          = 0.          UCPCE   (1)    = -23.069     UCPCE   (2)    =  2.5955        UCPCE   (3)    =  20.062
XCPCE   (1)    -  1378.6     XCPCE   (2)    = -35.009     XCPCE   (3)    =  30.287        ECPCE   (1)    = -706.72
```

Figure 9. (Continued)

```
ECPCE   (2)    =  44.081    ECPCE   (3)    =  724.48    XPCPCRC(1)    =  1364.2    XPCPCRC(2)    = -34.187
XPCPCRC(3)     =  15.779    XPCPCDC(1)     =  1095.7    XPCPCDC(2)    = -22.944    XPCPCDC(3)    =  18.742

        TIME =  4.900
UDST    (1)    =  35.757    UDST    (2)    =  891.94    UDST    (3)   = -52.239    USTSE   (1)   = -113.56
USTSE   (2)    =  .27114    USTSE   (3)    = -85.402    SRPSE   (1)   =  1580.4    SRPSE   (2)   = -41.904
SRPSE   (3)    =  75.057    WDST    (1)    = -119.38    WDST    (2)   =  418.84    WDST    (3)   = -368.13
WSTSE   (1)    =  34.852    WSTSE   (2)    =  18.053    WSTSE   (3)   =  367.88    ESTSE   (1)   =  185.10
ESTSE   (2)    = -.54925    ESTSE   (3)    =  889.60    RADAM         =  .10199    DREAM         =  .32205
CF CT          = 0.         UCPCE   (1)    = -4.1329    UCPCE   (2)   = -17.143    UCPCE   (3)   =  24.771
XCPCE   (1)    =  1376.3    XCPCE   (2)    = -34.929    XCPCE   (3)   =  39.362    ECPCE   (1)   = -766.76
ECPCE   (2)    =  2.0555    ECPCE   (3)    =  888.73    XPCPCRC(1)    =  1385.4    XPCPCRC(2)    = -34.323
XPCPCRC(3)     =  18.022    XPCPCDC(1)     =  1096.0    XPCPCDC(2)    = -22.851    XPCPCDC(3)    =  21.421

                              INTEGRATOR STEP SIZE LIMITING COUNTS


  1 EF CT    = 0.          2 EL CT    = 0.          3 WK CT    = 0.          4 WB CT   = 0.          5 WI SR   = 0.
UAPSL  ( 1) = 0.         UAPSL  ( 2) = 0.         UAPSL  ( 3) = 0.
XAPSL  ( 1) = 0.         XAPSL  ( 2) = 0.         XAPSL  ( 3) = 0.
WAPSL  ( 1) = 0.         WAPSL  ( 2) = 0.         WAPSL  ( 3) = 0.
EAPSL  ( 1) = 0.         EAPSL  ( 2) = 0.         EAPSL  ( 3) = 0.
 18 WG SP    = 0.
ESGSP  ( 1) = 0.         ESGSP  ( 2) = 0.         ESGSP  ( 3) = 0.
ESRSP  ( 1) = 0.         ESRSP  ( 2) = 0.         ESRSP  ( 3) = 0.
 25 EF MP    = 0.         26 EL MP    = 0.         27 WK MP    = 0.          28 WB MP   = 0.          29 EC LIRC = 0.
 30 TF LIRC  = 0.
UPPPCRC( 1) = 0.         UPPPCRC( 2) = 0.         UPPPCRC( 3) = 0.
XPPPCRC( 1) = 0.         XPPPCRC( 2) = 0.         XPPPCRC( 3) = 0.
WPPPCRC( 1) = 0.         WPPPCRC( 2) = 0.         WPPPCRC( 3) = 0.
EPPPCRC( 1) = 0.         EPPPCRC( 2) = 0.         EPPPCRC( 3) = 0.
UPCPCRC( 1) = 0.         UPCPCRC( 2) = 0.         UPCPCRC( 3) = 0.
XPCPCRC( 1) = 0.         XPCPCRC( 2) = 0.         XPCPCRC( 3) = 0.
 49 EC LIDC  = 0.         50 TF LIDC  = 0.
UPPPCDC( 1) = 0          UPPPCDC( 2) = 0.         UPPPCDC( 3) = 0.
XPPPCDC( 1) = 0.         XPPPCDC( 2) = 0.         XPPPCDC( 3) = 0.
WPPPCDC( 1) = 0.         WPPPCDC( 2) = 0.         WPPPCDC( 3) = 0.
EPPPCDC( 1) = 0.         EPPPCDC( 2) = 0.         EPPPCDC( 3) = 0.
UPCPCDC( 1) = 0.         UPCPCDC( 2) = 0.         UPCPCDC( 3) = 0.
XPCPCDC( 1) = 0.         XPCPCDC( 2) = 0.         XPCPCDC( 3) = 0.
UCPCE  ( 1) = 0.         UCPCE  ( 2) = 0.         UCPCE  ( 3) = 0.
XCPCE  ( 1) = 0.         XCPCE  ( 2) = 0.         XCPCE  ( 3) = 0.
WCPCE  ( 1) = 0.         WCPCE  ( 2) = 0.         WCPCE  ( 3) = 0.
ECPCE  ( 1) = 0.         ECPCE  ( 2) = 0.         ECPCE  ( 3) = 0.
 81 SCDCE    = 0.         82 SC CE    = 0.
UABABSK( 1) = 0.         UABABSK( 2) = 0.         UABABSK( 3) = 0.
XABABSK( 1) = 0.         XABABSK( 2) = 0.         XABABSK( 3) = 0.
WABABSK( 1) = 0.         WABABSK( 2) = 0.         WABABSK( 3) = 0.
EABABSK( 1) = 0.         EABABSK( 2) = 0.         EABABSK( 3) = 0.
USTSE  ( 1) = 0.         USTSE  ( 2) = 0.         USTSE  ( 3) = 0.
SRPSE  ( 1) = 0.         SRPSE  ( 2) = 0.         SRPSE  ( 3) = 0.
WSTSE  ( 1) = 0.         WSTSE  ( 2) = 0.         WSTSE  ( 3) =  2.0000
ESTSE  ( 1) = 0.         ESTSE  ( 2) = 0.         ESTSE  ( 3) = 0.
107 SCDSE    = 0.        108 SC SE    = 0.

        TIME =  5.000
UDST    (1)    =  377.62    UDST    (2)    =  455.15    UDST    (3)   = -79.786    USTSE   (1)   = -90.013
USTSE   (2)    =  59.158    USTSE   (3)    = -73.747    SRPSE   (1)   =  1571.8    SRPSE   (2)   = -42.048
SRPSE   (3)    =  81.487    WDST    (1)    = -608.27    WDST    (2)   =  86.844    WDST    (3)   = -860.41
WSTSE   (1)    = -2.0569    WSTSE   (2)    =  41.181    WSTSE   (3)   =  296.31    ESTSE   (1)   =  152.14
ESTSE   (2)    = -8.4608    ESTSE   (3)    =  894.18    RADAM         =  .13124    DREAM         = -1.1801
```

Figure 9. (Continued)

```
CF CT         = 0.         UCPCE  (1)   =  13.873    UCPCE  (2)   = -5.0871    UCPCE  (3)   =  25.901
XCPCE  (1)    =  1375.9    XCPCE  (2)   = -34.839    XCPCE  (3)   =  42.336    ECPCE  (1)   = -827.40
ECPCE  (2)    = -28.725    ECPCE  (3)   =  718.79    XPCPCRC(1)   =  1386.8    XPCPCRC(2)   = -34.506
XPCPCRC(3)    =  20.484    XPCPCDC(1)   =  1096.2    XPCPCDC(2)   = -22.957    XPCPCDC(3)   =  24.106


     385.456     CPU SECONDS WERE REQUIRED FOR THE PREVIOUS ANALYSIS
```

Figure 9. (Continued)

HUMAN TOLERANCE ANALYSIS THROUGH   3.942 SECONDS OF THE SIMULATION

AEROMEDICAL SIGN CONVENTION ......

GX = +ACCEL, GY = +ACCEL, GZ = -ACCEL ......

GXMAX =	7.71	TIME =	3.009
GYMAX =	9.25	TIME =	1.245
GZMAX =	20.83	TIME =	2.887
GXMIN =	-21.41	TIME =	.589
GYMIN =	-32.03	TIME =	.510
GZMIN =	-15.81	TIME =	.442
DRIMAX =	24.02	TIME =	3.024
RADMAX =	2.55	TIME =	.510

FIGURES OF MERIT.................

EXPERIENCE FACTOR - TOTAL LOAD =	.770
EXPERIENCE FACTOR - SAFE LOAD =	.703
EXPERIENCE FACTOR - UNSAFE LOAD =	.067

Figure 9. (Continued)

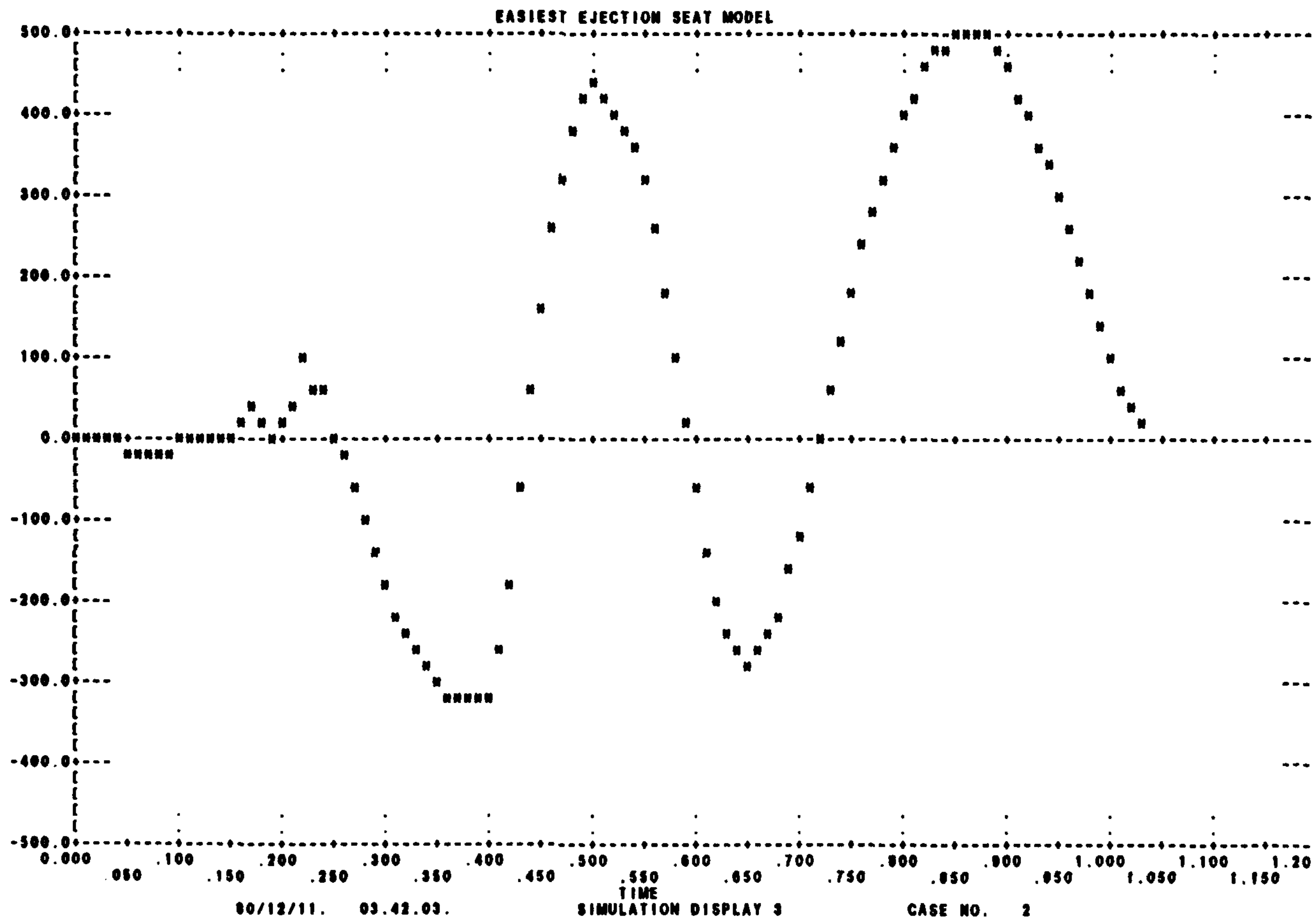


Figure 10. Example of Printer Plots

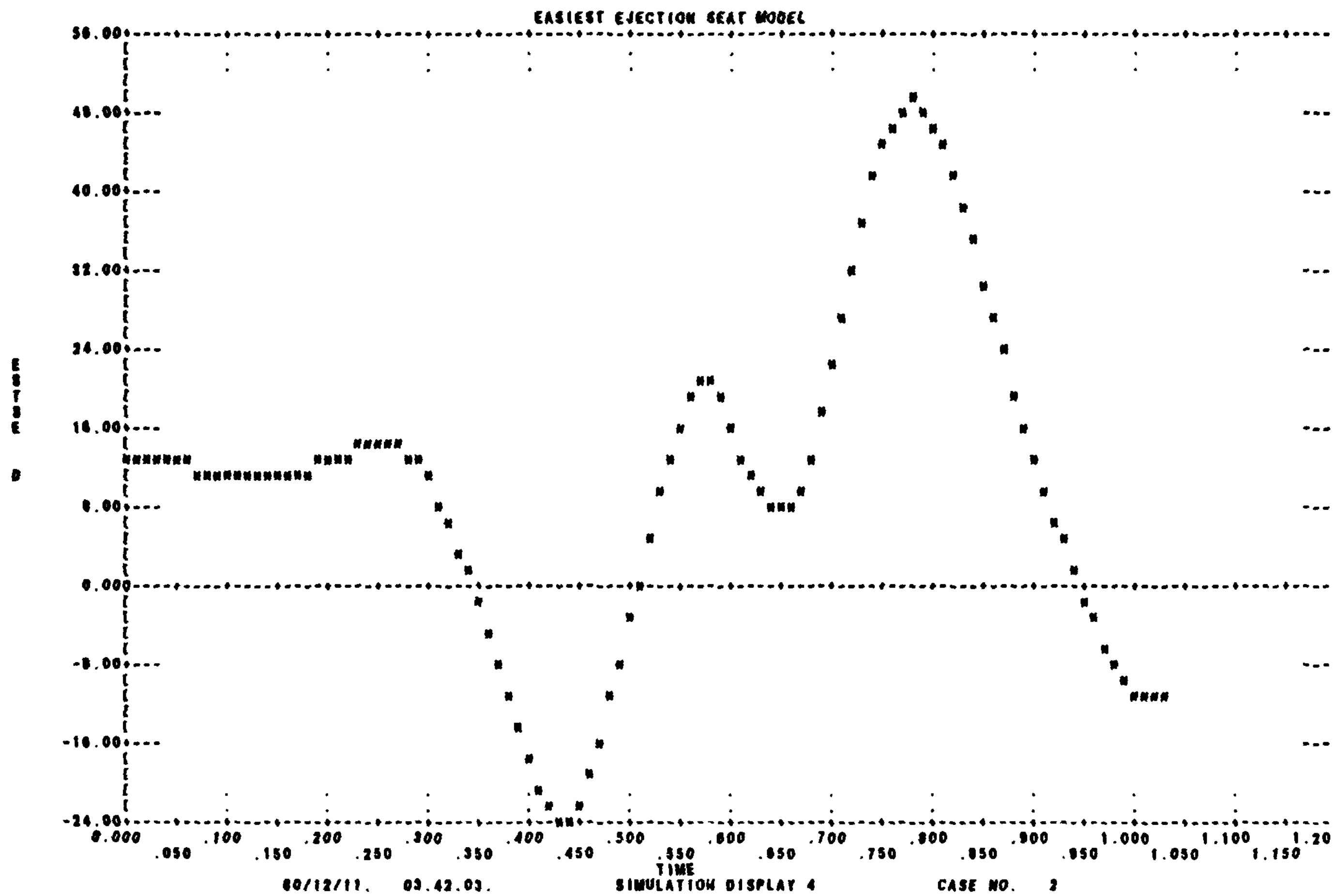


Figure 10. (Continued)

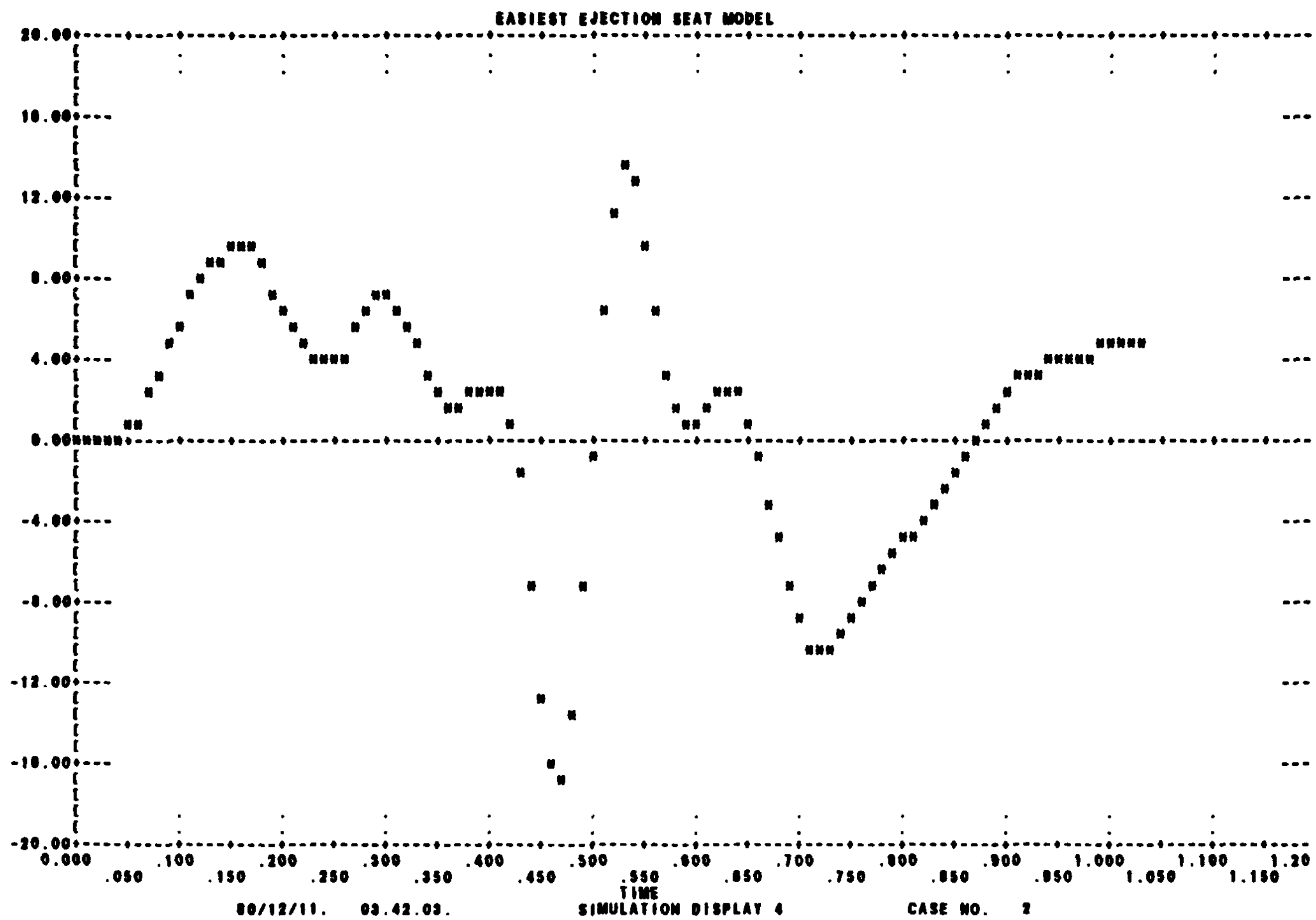


Figure 10. (Continued)

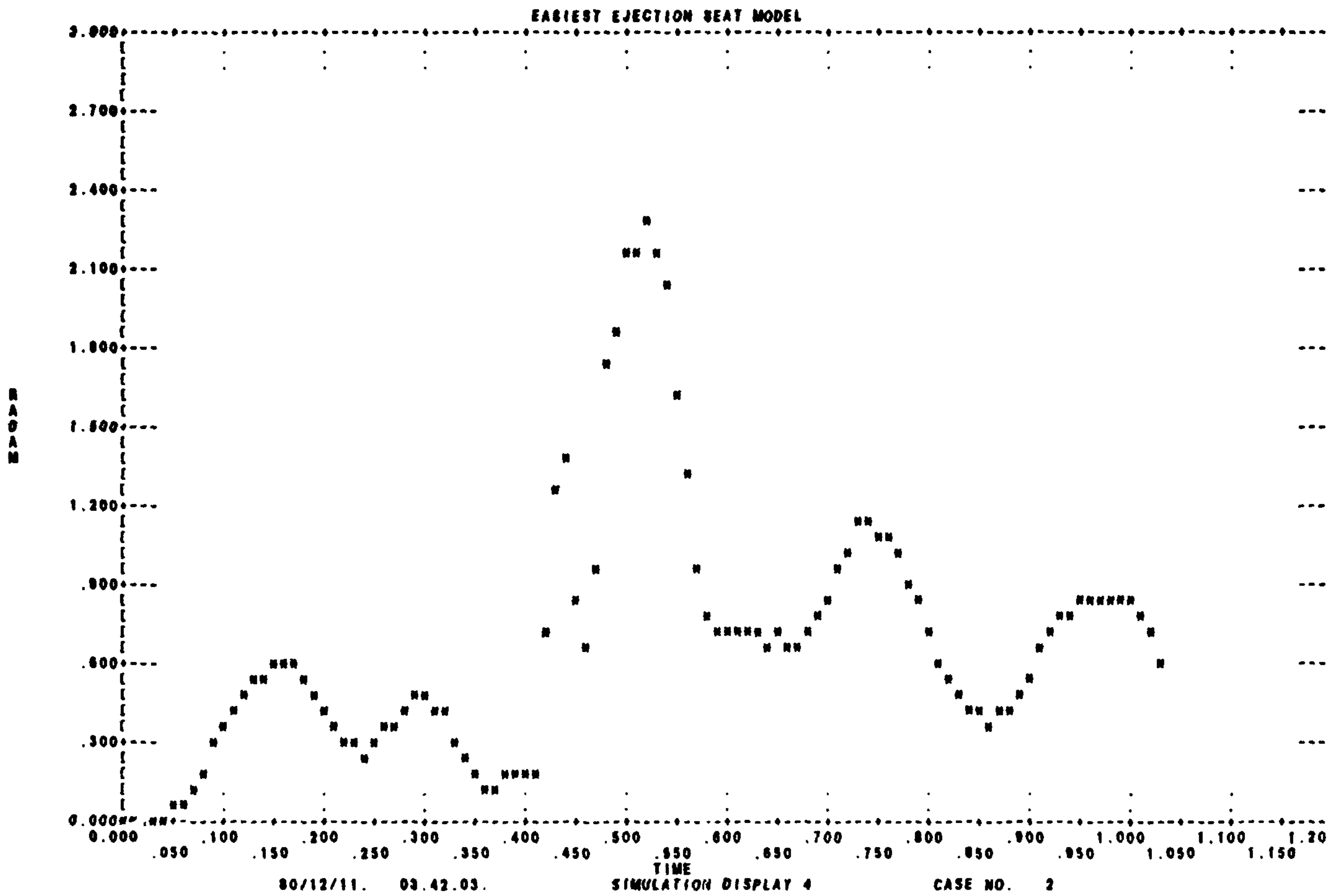


Figure 10. (Continued)

SECTION VII
CREATING AND MODIFYING STANDARD COMPONENTS

EASIEST is a collection of components designed to be used with the EASY Dynamic Analysis System (EASY), and any additions or modifications to the EASIEST components should take this into account. Before we describe the steps required to modify EASIEST, we give a brief introduction to the structure of EASY emphasizing the constraints that the structure puts on components used with it.

EASY consists of the EASY Model Generation Program and the EASY Analysis Program, plus various routines, files, and procedures to maintain and execute these two programs. EASY uses the EASY Model Generation Program to convert the user's model description file into a FORTRAN subroutine called EQMO. Each time a standard component is specified in the user's model a call is generated in EQMO to the subroutine with the same two character name as the standard component name. The EASY Model Generation Program generates these calls using data contained in a random access file called EZSTDBF. This file contains the names and specificatons of all inputs, outputs, and tables for each component. EQMO is used by the EASY Analysis Program under direction of the user's analysis file in the following way: given the value of time and the values of all the state variables in the model, compute the rates at which the state variables would be changing at that time. Note that the values of the state variables are not computed by EQMO or its subroutine (including standard component subroutines). The states are computed by the EASY Analysis Program using the rate data provided by EQMO. The rate data is used in different ways during different analyses (simulation, steady state, linear analysis, etc.). Therefore we have:

> Constraint #1. The user should make no changes to a standard component subroutine or any subroutine called by a standard component subroutine which results in a value to be assigned to a state variable.

Exception: during the CALC XIC analysis the EASY Analysis Program expects EQMO to compute state variables. Examples of how to set this up can be seen by examining the FORTRAN code of some of the existing standard components.

Each standard component consists of a FORTRAN subroutine in the EASIEST library with the same name as the two character standard component name and three records on a random access permanent file EZSTDBF. If QZ was an EASIEST standard component, then EZSTDBF would contain records called QZINPT, QZOUTP, and QZTABS. QBTABS contains one word for each table used by the component plus one word containing the number of tables. QZOUTP contains one word for each of the component's output quantities plus one word containing the nunber of output quantities. QZINPT contains one word for each input quantity (excluding tables) and one word containing the number of input quantities. These records are used by the EASY Model Generation Program to construct the calls in EQMO to the standard component subroutines. The calling sequences are constructed in the following order: one entry for each table, followed by one entry for each output quantity, followed by one entry for each input quantity. The exception is that an output quantity of a component which is declared to be a state variable will have only one entry in EZSTDBF, but will have three entries in the calling sequence of the component's FORTRAN SUBROUTINE, one for the value of the state variable itself (typed real), one for the value of the rate of the state variable (real), and one for a integration control flag (integer), in that order. The order of the calling sequence generated must correspond to the order of quantities in the SUBROUTINE card in the standard component subroutine. Therefore we have:

Constraint #2: Every change effecting the SUBROUTINE card of the standard component subroutine must be accompanied by a corresponding change in the component's records in EZSTDBF, and visa versa.

The steps required to modify EASIEST depend upon the type of modification being made. Each type is discussed below.

1. MODIFYING THE FORTRAN SUBROUTINE OF AN EXISTING STANDARD COMPONENT

The FORTRAN source code for the EASIEST standard components is stored on permanent file EZSTFTN (SN=AFFDL,no passwords). This file also contains source code of routines used by the standard component routines. The contents of this file can be cataloged by editing the file with INTERCOM EDITOR and typing:

L,A,/SUBROUT/

The listing produced on the terminal will be called the "catalog listing". Note that function subroutines do not appear in the catalog. They are located at the end of EZSTFIN, and should not affect the modifying procedures. Each subroutine in the listing resides on a separate record of EZSTFTN and you should note the record number of the subroutine you wish to change. Also, the line numbers on the catalog listing can be used in conjunction with the line numbers on the current FTN output listing to locate the line(s) of EZSTFTN to be changed.

Once the changes have been made, the edit file should be saved and cataloged as a new cycle of EZSTFTN, and the previous cycle should be purged from the disk. The EASIEST library EZSTLIB must now be updated to reflect the changes made to the source code. To do this attach file EZSTPRC (SN=AFFDL, PW=PSWD) and type:

BEGIN,COMPILE,EXSTPRC,n,CODE=cc

where:

1. n is the record number of the record on EZSTFTN you changed (this number can be obtained by counting down on the catalog listing described above),
2. cc is a two character code used in the output listing filename,
3. tid is identifier of the terminal into whose print queue you wish the FTN output listing placed (this entry is required only if you wish the output listing directed to a terminal other than the default terminal AB).

A successful execution of this procedure means that EZSTFTN has now been updated to reflect your change. If the FTN compiler does not accept the changes you made to EZSTFTN, the COMPILE procedure will leave the EASIEST library unchanged and make the FTN output listing containing the error

description available as local file FTNLIST. This file can be examined from the terminal using the INTERCOM EDITOR or PAGE utilities. When the trouble is located, correct EZSTFTN and rerun the compile procedure as described above.

2. MODIFYING THE RANDOM ACCESS FILE EZSTDBF.

If changes are made to a standard component subroutine involving either the number or characteristics of the components inputs, outputs, or tables, then in addition to the steps given in section VII.1 for altering the component's FORTRAN subroutine, the component's EZSTBDF records must be altered so that the EASY Model Generation Program will alter the generated calling sequences for the component. EZSTDBF is altered using a program called FILOAD which in turn is executed from an INTERCOM terminal using a procedure called DBFMOD contained on the procedure file EZSTPRC. DBFMOD requires the user to supply a permanent file containing all the data to build the record or records being modified. This file can have any otherwise unused name; for illustrative purposes we will assume it is called DBFDATA. For each record of EZSTDBF being modified the file DBFDATA must contain the following data:

i. A line describing the number on entries in the record in the form:

"xxINPUTS=n", or "xxOUTPS=n", or "xxTABS=n"

where xx is the component name and n is the number of inputs, outputs, or tables.

ii. One or more lines containing the names and specifications of the inputs, outputs, or tables for the component. Each of these lines (except possibly the last) must contain entries for eight quantities. Each entry consists of exactly ten characters including spaces and must begin in columns 1,11,21,31,41,51,61,71, or 81 of the line. These entries must be placed eight to a line until the specified number of quantities has been given. Each of the entries has the following format:

Character	Contents
1-3	the quantity name (inputs, outputs, or tables)
5-6	the quantity row dimension, if any (inputs,outputs)
7-8	the quantity column dimension if any (inputs,outputs)
9	the quantity port number if any (inputs,outputs)
10	=S if a state (outputs only)
?-?	total storage allocation (tables only)
?-?	number of independent variables (tables only)

The dimensions can be one or two digit numbers or can be the symbols N or M which allows the dimensions of the quantity to be set in the model description file. If any input or output quantity of a component is to have variable dimensions, the DBFDATA file should also have a seperate line of the form:

MODES = xx

where xx is the component name.

If more than one record of EZSTDBF is to be modified, the input data for each re ord can be placed on successive lines of DBFDATA.
An easy way to generate the file DBFDATA is to have the procedure DBFMOD generate a local file TMPDATA which contains all the input data to build EZSTDBF as it is now. To do this:

1. Create a permanent file DUMPFIL (no password) containing "DUMP FILE" on a single line of text.
2. Attach the procedure file EZSTPRC;
3. While in INTERCOM command mode type:
   BEGIN,DBFMOD, EZSTPRC,DUMPFIL,EZSTDBF,TMPDATA

Upon successful completion of DBFMOD you will have a local file TMPDATA containing all the input data required to generate the current version of EZSTDBF. The file DUMPFIL can now be purged. Using the INTERCOM EDITOR utility, delete all lines of TMPDATA pertaining to records of EZSTDBF not being modified, make the desired changes to the remaining lines, and save the edit file as DBFDATA.

Once you have the revised DBFMOD input data prepared on file DBFDATA and have cataloged DBFDATA on your account (with no password), attach the file EZSTPRC as before if you have returned it, and type:

BEGIN,DBFMOD,EZSTPRC,DBFDATA,EZSTDBF

Upon successful completion of this procedure, EZSTDBF will have been updated. You may now purge the file DBFDATA. It is recommended that the model description file of the next EASIEST run you submit contain the first line

LIST STANDARD COMPONENTS

This will cause the lineprinter output from that run to contain a listing of all the input, output, and table data for all the standard components. From this listing you can verify that the desired changes have been made to EZSTDBF.

3. CREATING A NEW EASIEST STANDARD COMPONENT

Creating a new component for EASIEST consists of constructing the source FORTRAN code, merging that code into the EASIEST library and constructing the input, output, and table descriptions for the random access file EZSTDBF. The FORTRAN source code for the new component subroutine and any new subroutines needed by your component subroutine should be prepared on a separate file following the constraints above. This code can then be merged into the EASIEST source as follows:

1. Attach the file EZSTFTN and, using INTERCOM EDITOR utility, obtain a "catalog listing' of EZSTFTN as described in section VII.1. Determine the proper position for your new subroutine so that the "Catalog listing" will remain alphabetical.
2. Request a permanent file PF by typing:

    REQUEST,PF,*PF

3. Copy the subroutines that are to preceed the new subroutine on EZSTFTN onto the file PF by typing:

    COPYCR,EZSTFTN,PF,n

   where n is the number of subroutine to preceed the new one. n can be obtained by counting down on the "catalog listing" of EZSTFTN.
4. Copy the source code of the new subroutine onto PF using

    COPY,f,PF

   where f is the name of the file containing the new source code. Note that file f must be attached before you do the copy.
5. Copy all the remaining subroutines from EZSTFTN onto PF using

    COPYCR,EZSTFTN,pf,999

   The terminal will respond with the number of records copied. This number should be checked against the "catalog listing" to make sure that all the subroutines have been copied. As added insurance, use the INTERCOM EDITOR utility to make a "catalog listing" of file PF and check that PF has the expected structure.
6. Catalog PF as new cycle of EZSTFTN using

    CATALOG,PF,EZSTFTN,RP=999

7. Purge the previous high cycle of EZSTFTN

PURGE,EZSTFTN

RETURN,EZSTFTN,PF

8. The new routine can now be compiled and merged into EZSTLIB using the procedure COMPILE as described in section VII.1. If more than one subroutine is to be added, repeat the above steps.

To include the input, output, and table data for the new component into EZSTDBF, create a permanent file DBFDATA as described in section VII.2. Usually you will have to supply data for three EZSTDBF records, xxINPT, xxOUTP, and xxTABS, where xx is the new component name. However, if the new component has no quantities of a certain type (inputs, outputs, or tables), then no input data of that type need be given. When the file DBFDATA is prepared and cataloged (no password), you can execute the procedure DBFMOD by typing:

ATTACH,EZSTPRC.

BEGIN,DBFMOD,EZSTPRC,DBFDATA,EZSTDBF

The terminal will type (among other things):

xx WILL BE ADDED AS A NEW STANDARD COMPONENT

You should include LIST STANDARD COMPONENTS command in the model description file of your next EASIEST run to verify that the inputs, outputs, and tables have been specified correctly.

4. LIBRARY EZSTLIB SIZE REDUCTION

Every time the procedure COMPILE is execute, the EASIEST library file EZSTLIB will grow in size. When this size becomes unreasonable EZSTLIB should be rebuilt anew from the source file EZSTFTN by typing the following sequence from an INTERCOM terminal in command mode:

ATTACH,EZSTPRC.

BEGIN,COMPALL, EZSTPRC,EZSTFIN,EZSTLIB

The successful completion of this procedure will mean that a new (smaller) cycle of EZSTLIB has been cataloged. The previous high cycle can then be deleted. The FORTRAN output listing from the FTN compilation phase is left available for routing to a lineprinter as local file ALLLIST.

SECTION VIII

DESCRIPTION AND GUIDE TO USE OF NUMERICAL INTEGRATION

The purpose of this section is: (1) to document changes (as they relate to the user) in integration methods used in the EASY program; (2) to describe local error control procedures in the three automatic integrators - NRKVS, STIFF GEAR, and ADAMS; and (3) to discuss the appropriate use of each method.

1. CHANGES IN INTEGRATORS

Several inadequacies in the integrators used in early versions of EASY were identified and subsequently remedied in the EASY5 program. In particular, the error control technique in the NRKVS integrator was reworked and the Hindmarsh version of C. W. Gear's integrator was implemented. The Hindmarsh version, called GEAR, also includes minor changes, such as dynamic dimensioning and the capability to input EASY5 error controls.

The resulting set of improved integrators are accessed by the EASY5 user through the integration method parameter, INT MODE. INT MODE can be set to any integer from 1 to 6 with the default being 6. The six integrators which are available are listed below.

a. DIFSUB: The original version of Gear's method.
b. NRKVS: The improved Runge-Kutta variable step integrator.
c. HEUNS: Second order fixed step explicit method.
d. Euler: First order fixed step explicit method.
e. ADAMS: Automatic step-size/order selection methods using Adams-Bashforth predictor/Adams-Moulton corrector pairs of (2nd through 12th) order. (Non-stiff option of GEAR.)
f. STIFF GEAR: The stiffly stable GEAR formulas.

The choice of the best integration method depends on a number of considerations. User requirements, problem characteristics, and the stability and accuracy of the method all must be considered. A more complete discussion of these considerations can be found in standard texts. It is the purpose of this section to present summary information to help the user with his integrator selection. The second and third sections discuss accuracy, error control, and stability in more detail and can be consulted if integration problems develop or simply to gain a better understanding of the processes involved.

2. GENERAL SELECTION GUIDELINES

Many times the best and only way to choose a method is by trial and error. Below are some general observations:

a. If no special knowledge is available about the system, try Method 5: ADAMS.

b. If a large amount of output is desired at small time increments, Methods 5 or 6 will use interpolation rather than generate smaller time steps if output points are smaller than current step sizes. However frequent restarting will cause the cost of an entire transient simulation to increase.

c. If function evaluation can only be calculated at fixed time steps due to sampling data or tabular information, use Methods 3 or 4. Method 3 is more efficient if the time step is obviously small enough to generate necessary accuracy. Given a fixed time step, Method 4 will be more accurate than method 3 provided h_0 is within the stability region (Figure 11) for the methods.

d. If the system has frequent derivative discontinuities (shocks, phase changes, hard step-like forces, etc) Method 2: NRKVS is recommended. Unlike Methods 5 and 6, Method 2 is negatively impacted by a large number of output points at small time increments (i.e., if the output

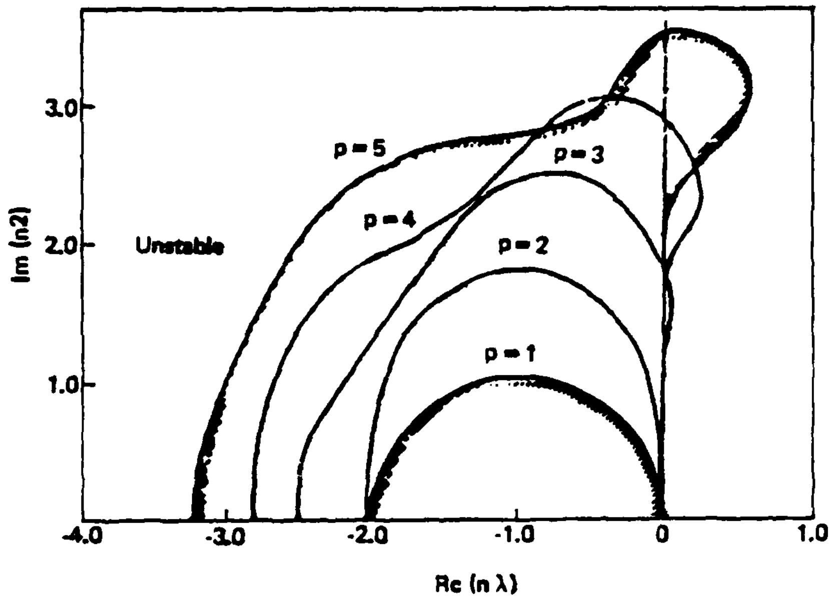


Figure 11. Stability Regions For Runge-Kutta Methods: Orders 1-5

time increments are smaller than the natural step size, then the method will be found to use more integration steps and thus be more costly).

e. Method 1 is not recommended. If the system is initially unstable or discontinuous and eventually is stiff, we recommend using Method 2, then switching to Method 6 rather than using Method 1.

f. If the problem is stiff (i.e., large spread in eigenvalues), Method 6: STIFF GEAR is recommended. This is also the default option if no method is specified.

It should be noted, however, that problems with large eigenvalues (with negative real parts) do <u>not</u> automatically indicate that one should use STIFF GEAR. For example, consider the system:

$$(1) \qquad \begin{aligned} X_1 &= -X_1 \\ X_2 &= -1000X_2 \end{aligned} \qquad \text{for time } 0 \leq t \leq b$$

This is an uncoupled system (and might seem artificial), but coupled systems often display the behavior of rapidly damping components such as X_2. If one was integrating (1) as a system and the important variable was X_1 and b was large, then a large step size could be used provided the numerical integration of X_2 was damping to zero (i.e., stable). In such a case, a STIFF method would be appropriate. On the other hand, if b was small (e.g., b = 0.0001) and X_2 was the component of interest (where relative accuracy is important), then an efficient integrator of Adams type or perhaps a Runge-Kutta method would be appropriate. Thus, the decision to use STIFF GEAR or not depends on both the user requirements for accuracy and the eigenvalues of the system.

3. ACCURACY AND ERROR CONTROL

It is useful to establish notation and review some basic concepts. Consider the ordinary differential equation (ODE)

(2) $X(t) - f(t,X(t))$ with $a \leq t \leq b$

with the initial condition $X(a) = X_0$. Equation (2) is an initial value problem. The EASY5 program sets up and solves first order systems of such equations (i.e., equations of the form of equation (2) with X a vector and f a vector valued function). Initial value methods for integrating ODE's produce a sequence X_j of approximations to the solution X such that X_j $X(t_j)$ where $t_0 = a$ and $t_j - t_{j-1} + h_j$ for $j = 1,N$. The sequence h_j are called steps or step sizes. For Methods 3 and 4 (Heun's and Euler's methods), the step sizes are fixed throughout the integration and are set by the user through the parameter TINC. For the other methods, the step sizes are selected by the integration algorithm as the integration proceeds. These "adaptive" methods estimate the local truncation error at each step of the integration, accept or reject the approximation, and predict the next step size to be tried. Local truncation error can be loosely thought of as the error incurred during one-step of the integration process given that all previous approximates are exact. The order of a method is a crude measure of accuracy. A method is said to be of order p if it is exact for pth order polynomials. The adaptive EASY5 integrators (Methods 1, 2, 5, and 6) strive to keep the step size small enough to insure reasonable local error which in turn should produce a small global error. Whether or not the global error is indeed small will depend on both the problem and the stability of the method. (Stability is discussed in the next section.)

The adaptive integrators measure the local truncation error by comparing two estimates of the solution that theoretically differ in only high order terms from the Taylor's expansion of the solution over the current step. The details of how this is done in each method is not important here except as to how it relates to the EASY5 integration controls. The user is asked

to input an array of controls associated with each state of the system via the ERROR CONTROL command. The array, which we shall call ERROR(I), is a measure of significance of the corresponding Ith state of the system. To be precise ERROR(I) is a value below which the Ith state is in some sense considered negligible by integrators 1, 2, 5, and 6. There are two methods of error control employed by the four methods. Method 2, NRKVS, is described first, Error control in Methods 1, 5, and 6 are basically the same and will be discussed second.

In NRKVS, the initial step size H_0 is chosen as a function of TINC. To be precise H_0 = .01 * TINC.

Subsequent step sizes are selected on the basis of local error control estimates. There are a number of refinements in NRKVS that will not be discussed; however, the basic error control is governed by the following quantity, Q,

$$(3) \quad Q = \max_{(I)} \left[\frac{\mathrm{LTE}(1)}{\mathrm{ERROR}(I) + |X(I)| * \mathrm{ERROR}(I)} \right]$$

where LTE(I) is the local truncation error estimate for the Ith state of the solution as calculated by comparing a 4th order solution to a 5th order solution, X(I) is a recent history size measure of the Ith state (initially set to the initial value), and ERROR(I) is the user input error control. The integrator strives to make Q = 1. If $Q<1$, the step size on the next integration step is increased. If $Q>10$, the current step is rejected and a new smaller step size is calculated for another attempt. In order to interpret the effect of the input controls, ERROR(I), one need only set Q = 1 (the desired value for Q) and examine the relation (2) for the maximal choice of I. That is, for some I, if Q = 1, then

$$(4) \quad Q = 1 = \frac{\mathrm{LTE}(I)}{\mathrm{ERROR}(I) + |\overline{X}(I)| * \mathrm{ERROR}(I)}$$

Thus by rewriting (4) we have that

$$LTE(I) = ERROR(I) + \overline{X}(I) * ERROR(I).$$

i.e., the LTE is close to the ERROR + X*ERROR. If X(I) has been small, ERROR (I) dominates the right hand side of (4), and the error control is essentially absolute error. On the other hand, if X(I) is very large, X(I)*ERROR(I) will dominate; and thus relative error is controlled. As a rule of thumb, the user should input the level at which he considers the solution negligible (i.e., tolerably small enough to ignore). If the solution gets large, then $\log_{10}(ERROR)$ will roughly give the number of significant digits of accuracy (locally).

The use of input controls ERROR(I) differs for Methods 1, 5 and 6. A local truncation error LTE is computed by the integrator. The Euclidean error is controlled, i.e.,

$$\sum_{I=1}^{NEQ} \left(\frac{LTE(I)}{XMAX(I)} \right)^2$$

is required to be less than $(EPS)^2$ where NEQ is the number of equations, XMAX(I) is the maximum of the Ith component of X over the course of the integration. The user impacts this control by effecting the initialization of XMAX(I) and the choice of EPS. EPS is chosen as follows:

$$EPS = \underset{(I)}{MIN}\ (ERROR(I))$$

with the constraint that EPS $\leq$ .01. If ERROR(I) $<$ 1.E-12 for all I, then EPS is set to 1.E-4. The initialization of XMAX(I) is given by

```
XMAX(I) = ERROR(I) /EPS
IF (XMAX(I).EQ.0) XMAX(I) = 1.
```

The net effect of these initializations for EPS and the XMAX array result in the ERROR array being used in a similar manner to its use in NRKVS. For example, if EPS = .001 and ERROR = .001, then XMAX - 1.0 and error control is essentially absolute error until the solution X(I) exceeds 1. If X(I) grows the error processing will gradually become relative since XMAX is set equal to X whenever X exceeds it. If the solution grows to a maximum value, and then decays, the error control will be relative to that maximum.

The user must remember that EPS is set by the smallest ERROR(I). Thus, in a two component system, if ERROR(1) = .001, and ERROR(2) = 1.0, the resulting controls will be as follows:

EPS = .001; XMAX(1) = 1; XMAX(2) = 100.

Thus, if X(1) = X(2) = 0 initially the integrator considers values less than 0.001 negligible for X(1) and values less than 1.0 negligible for X(2). This is quite similar to what NRKVS would do with these same inputs for ERROR(1) and ERROR(2).

4. STABILITY

The theoretical basis for error control and convergence of numerical integration methods is rooted in the underlying assumption that the step size is small (in fact, approaching zero). In practice, of course, the step size is not necessarily small and certainly not zero. In fact, the larger the step size, the fewer the steps required, and hence, the greater the economy of integration. The behavior of integration methods when the step size gets large will generally depend on both the problem and the "stability" of the method. All the EASY5 integrators are at least "conditionally stable". That is, there exists a threshold size, h_0, such that for steps of $h < h_0$ the integration procedure will produce damping approximations to damping components. To be precise, consider the equation

$$(5) \qquad X(t) = \lambda X$$

where λ is any complex number. If λ has negative real part, the equation is said to be mathematically stable, and its solution may be oscillatory but definitely will damp with time. Given a method, one can calculate a stability region in the complex plane which depicts the region in the h plane for which the integration scheme will produce a damping solution to equation (5). That is, given a λ, the product $h\lambda$ must be within the absolute stability region for the method to produce a damping solution. Generally, if one uses a step size h outside this region for more than a few successive steps, numerical instability will occur producing a divergent "solution" even for a stable system. This, in fact, often happens with fixed step methods. Adaptive integrators will automatically reject these numbers and cut the step size, thereby increasing work (not because of accuracy) but because of stability. For systems of nonlinear differential equations, in equation (5) corresponds to the eigenvalues of the system. In Figure 11, the stability regions for Runge-Kutta methods of orders 1-5 are shown. The method will be stable provided $h\lambda$ is within these closed regions.

The region marked p=1 is valid for Euler's method (No. 4 in EASY5). Thus if, for example, $\lambda = -1000$, $h\lambda$ is required to be > -2 in order to produce meaningful results. This, in fact, implies that $h < .002$. The region p=2 corresponds to Heun's method which is Method 3 in EASY5. For $\lambda = -1000$, h must also be less than .002 for stability. In this case, since Method 4 uses only one function evaluation per step and Method 3 uses two, the Euler method would be more efficient on $\dot{X}(t) = -1000X$ if minimal accuracy were needed. On the other hand, if extremely accurate results were required, and the user intended to use small step sizes well within the stable regions, then the higher order accuracy of Heun's method would more than justify its extra function evaluation.

For a certain class of equations (stiff equations) the eigenvalues may vary considerably between components. For example, consider

$$\dot{X}_1 = -1000\ X_1$$
$$\dot{X}_2 = -X_2$$

The user often demands greater accuracy in X_2 than in X_1. Assume for the moment these equations are coupled. Then X_1 drives the step size used for the system. It is in this situation that large stability regions are desirable, for then a large step size can be used.

The p = 4 and p = 5 regions in Figure 11 are then stability regions that apply to the Runge-Kutta method NRKVS. The underlying method is 4th order in NRKVS; however, the error control mechanism performs an extrapolation to achieve a fifth order approximation to the computed 4th order estimate. The difference between the two is then used to estimate the error. The reported solution is the fifth order estimate; hence p - 5 is the true region of interest.

The Adams-Moulton formulas, Method 5, have stability regions given in Figure 12 for orders 3-6. (The Adams-Moulton methods are the corrector of the predictor-corrector pairs used in Method 5.) A corrector formula is implicit and if interated to convergence, will have the stability shown in Figure 12. However, the implementation of the Adams formulas in this code (and most Adams' codes) uses prediction with only one correction. The resulting stability regions are reduced. A sample of these regions for orders 1, 2, 3, 4, 5, 6, 9, and 10 are given in Figure 13a through h. The solid lines are for the Adams-Bashforth predictors; the dotted lines are for the Adams-Bashforth predict with the Adams-Moulton corrector of the same order (one correction); and the dashed lines are order k predict/order k+1 correct. The dotted lines represent the actual implementation in EASY5.

Thus far, the stability regions discussed have all been finite (bounded) regions of the plane. Consequently, to remain in the stability region of the plane for any λ with very large absolute value, one must use a very small step size. The advantage of the STIFF GEAR formulas (Method 6 in EASY5) is that their stability regions have infinite extent. This is shown graphically in Figures 14 and 15. For orders K=1,2, these methods are A-stable which means that for any λ with negative real part λh will fall within the stable region for any $h > 0$. The higher order formulas (Figure 15) impose restrictions on the size of the imaginary part of that

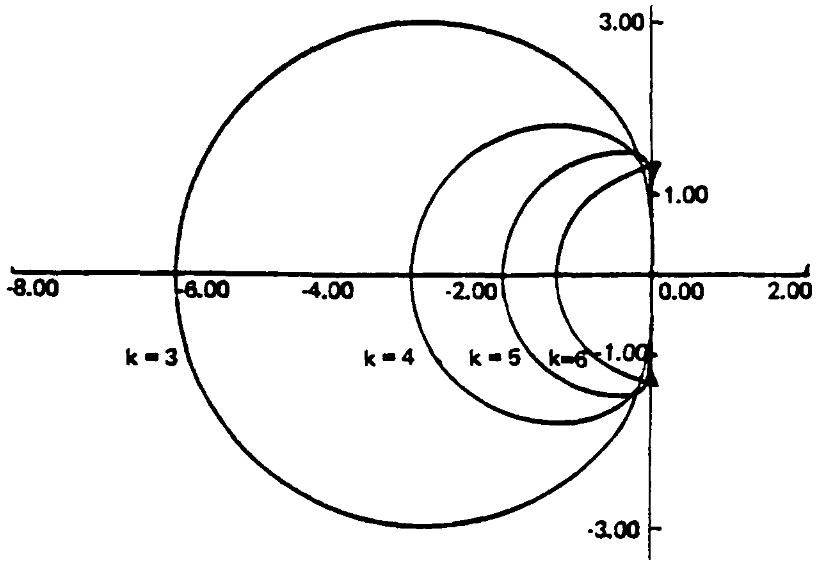


Figure 12. Stability Regions For Adams-Moulton Formulas

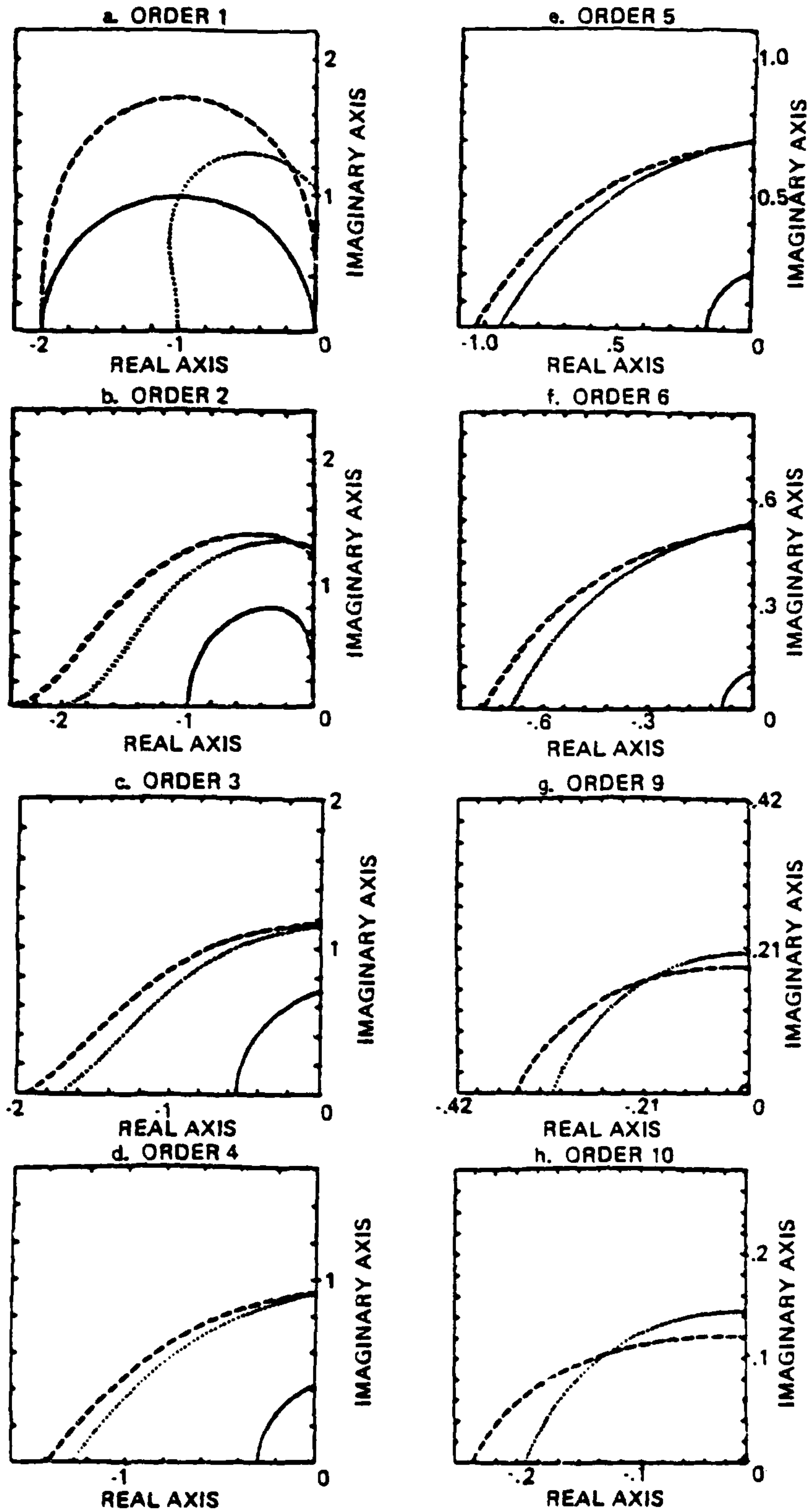


Figure 13a thru h. Stability Regions for Predictor-Corrector Pairs

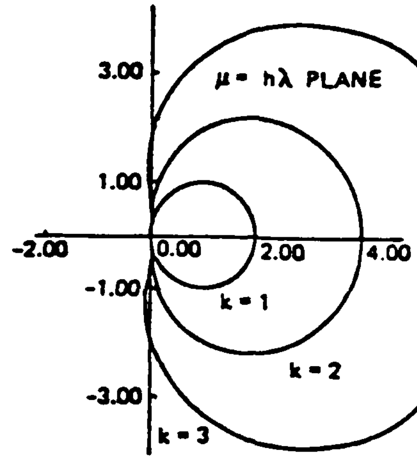


Figure 14. Stability Regions for STIFF GEAR Formulas of Orders 1-3

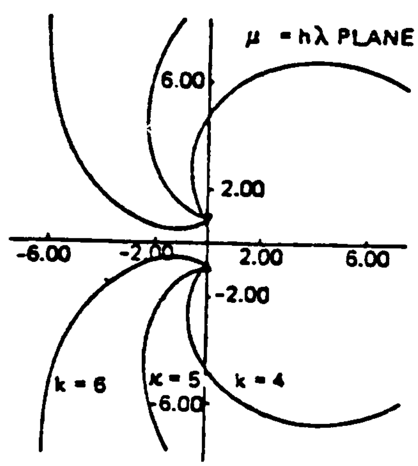


Figure 15. Stability Regions for STIFF GEAR Formulas of Orders 4-6

allow for large values of h. For example, the sixth order formula requires small h if λ is -1000 + 1000i, but for λ = -1000 the sixth order formula is stable for all values of $h > 0$. In the EASY5 program, implementation of STIFF GEAR the order may vary from one to five. Since Methods 1, 5, 6 are variable order codes, they will range over various orders during the integration. If the eigenvalues are close to the imaginary axis, Method 6 will probably use only orders 1, 2, and possibly 3 if it is constrained by stability. These highly stable methods generally require more function evaluations than the other methods mentioned (due to internal approximations to the Jacobian of the system required to solve implicit equations).

SECTION IX
DISCRETE SYSTEM ANALYSIS TECHNIQUES

1. INTRODUCTION

The discrete system analyses of the EASY program are based on the state space approach described by Kalman and Bertram in Reference 1. The EASY analyses utilize the single and multirate sampling capabilities of the original analysis. Other capabilities such as the analysis of nonsynchronous, noninstantaneous, multiple order, and random sampling are not currently implemented in the EASY program. The EASY program analyses parallel those of the M-DELTA program. However, whereas the M-DELTA program requires the user to input the $\underline{A}$ and $\underline{B}$ matrices that described the system, the EASY program calculates these matrices from a nonlinear system model described in terms of standard modeling components.

Only the linear analyses of the EASY program utilize the techniques of Kalman and Bertram. Since only the eigenvalues of the system are used in these analyses, the system equations will be simplified in the following derivations by treating the system as autonomous.

2. SYSTEM EQUATIONS

A discrete system may be described by the following three types of states:

a. Continuous States
b. Delay States
c. Sample and Hold States

The continuous states may vary continuously as a function of time and are each defined by a first order ordinary differential equation. Delay states are defined at only discrete points in time by first order difference equations. Sample and hold states maintain constant values except at discrete points in time where they may jump to new values. Figure 16 shows an example of each state type.

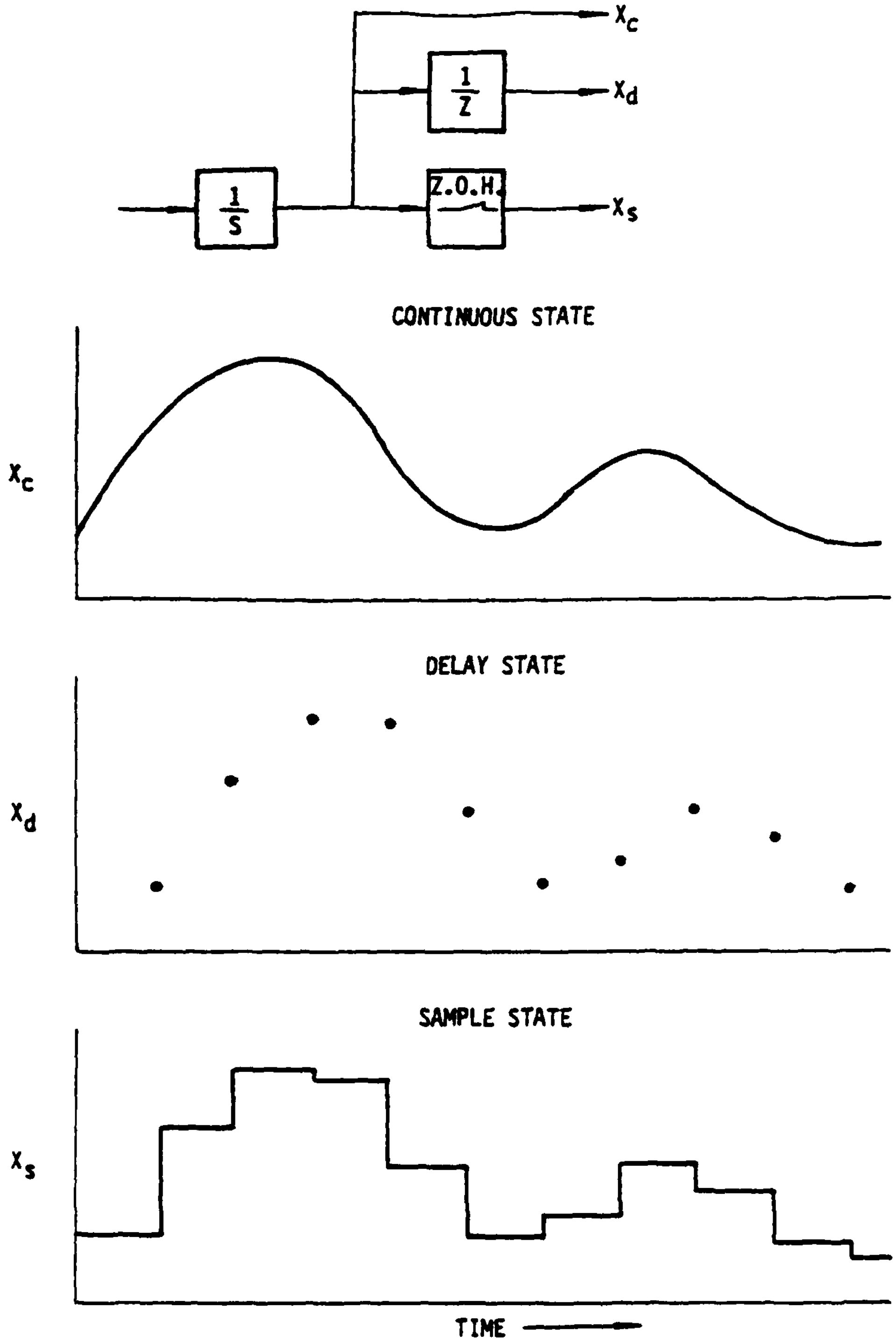


Figure 16. Example of Continuous, Delay, and Sample States

Let the continuous delay and sample state be grouped together as three state vectors:

$\underline{X}_c$ $\quad \gamma$ VECTOR OF CONTINUOUS STATES

$\underline{X}_d$ $\quad \delta$ VECTOR OF DELAY STATES

$\underline{X}_s$ $\quad \sigma$ VECTOR OF SAMPLE STATES

The total system state vector of dimension $\gamma + \delta + \sigma$ is formed into the single partitioned vector:

$$\underline{X} = \begin{bmatrix} \underline{X}_c \\ \hdashline \underline{X}_d \\ \hdashline \underline{X}_s \end{bmatrix} \tag{1}$$

a. Continuous System Stability Matrix

Between sample instants, the autonomous system behavior is described by:

$$\dot{\underline{X}} = \underline{AX} \tag{2}$$

The system stability matrix $\underline{A}$ between sampling instants may be expressed as the partitioned matrix:

$$\underline{A} = \begin{bmatrix} \underline{A}_{cc} & \underline{0} & \underline{A}_{cs} \\ \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} \end{bmatrix} \tag{3}$$

The form of the system stability matrix demnstrates that only the continuous states have non-zero rates, i.e., can change between sampling

instants, and that the continuous state rates are functions of only the continuous states and sample states.

b. Discrete System Transition Matrix

At sampling instants, the system behavior is described by:

$$\underline{X}(t+) = \underline{B}\,\underline{X}(t-) \tag{4}$$

For a single rate sampling system, the discrete transition matrix $\underline{B}$ will be of the form:

$$\underline{B} = \left[\begin{array}{c|c|c} \underline{I} & \underline{0} & \underline{0} \\ \hline \underline{B}_{dc} & \underline{B}_{dd} & \underline{0} \\ \hline \underline{B}_{sc} & \underline{B}_{sd} & \underline{0} \end{array}\right] \tag{5}$$

The form of the transition matrix at sampling instants demonstrates that the continuous states remain unchanged, i.e., the upper Υ rows contain only an identity matrix. The discrete states are functions of only the continuous and delay states at the previous sample instant.

c. Continuous System Transition Matrix

Equation (4) describes the instantaneous changes that occur in the system at sample instants while equation (2) describes the system between sampling instants. In order to combine these two types of behavior, we will convert the continuous description of (2) into a transition matrix that describes the transition between two sample instants.

Expanding (2):

$$\begin{aligned}
\dot{\underline{X}}_c &= \underline{A}_{cc}\,\underline{X}_c + \underline{A}_{cs}\,\underline{X}_s \\
\dot{\underline{X}}_d &= \underline{0} \\
\dot{\underline{X}}_s &= \underline{0}
\end{aligned} \tag{6}$$

Take Laplace transform

$$\begin{aligned}
s\underline{X}_c(s) - \underline{X}_c(0) &= \underline{A}_{cc}\,\underline{X}_c(s) + \underline{A}_{cs}\,\underline{X}_s(s) \\
s\underline{X}_d(s) - \underline{X}_d(0) &= \underline{0} \\
s\underline{X}_s(s) - \underline{X}_s(0) &= \underline{0}
\end{aligned} \tag{7}$$

Rearrange terms to solve for $X_c(s)$, $X_d(s)$, and $X_s(s)$:

$$\begin{aligned}
\underline{X}_c(s) &= \left[s\underline{I} - \underline{A}_{cc}\right]^{-1}\underline{X}_c(0) + \frac{\left[s\underline{I} - \underline{A}_{cc}\right]^{-1}}{s}\,\underline{A}_{cs}\,\underline{X}_s(0) \\
\underline{X}_d(s) &= \frac{\underline{X}_d(0)}{s} \\
\underline{X}_s(s) &= \frac{\underline{X}_s(0)}{s}
\end{aligned} \tag{8}$$

Take inverse Laplace transform:

$$\begin{aligned}
\underline{X}_c(\tau) &= e^{\underline{A}_{cc}\tau}\,\underline{X}_c(0) + \underline{A}_{cc}^{-1}\left[e^{\underline{A}_{cc}\tau} - \underline{I}\right]\underline{A}_{cs}\,\underline{X}_s(0) \\
\underline{X}_d(\tau) &= \underline{X}_d(0) \\
\underline{X}_s(\tau) &= \underline{X}_s(0)
\end{aligned} \tag{9}$$

Equation (9) is in the form of a transition equation from an initial time to a final time τ . It is also of the same form as equation (4) and may be written as:

$$\underline{X}(\tau) = \underline{\Phi}(\tau)\ \underline{X}(0) \tag{10}$$

Where:

$$\underline{\Phi}(\tau) = \left[\begin{array}{c:c:c} e^{\underline{A}_{cc}\tau} & \underline{0} & \underline{A}_{cc}^{-1}\left[e^{\underline{A}_{cc}\tau} - \underline{I}\right]\underline{A}_{cs} \\ \hdashline \underline{0} & \underline{I} & \underline{0} \\ \hdashline \underline{0} & \underline{0} & \underline{I} \end{array}\right] \tag{11}$$

When written in this form, we see that the transition matrix of the system between sampling instants is composed of the exponential decay term $e^{\underline{A}_{cc}\tau}$ due to the continuous states plus the effect of the step input from the sample states. The discrete states are constant between sampling instants as evidenced by the identity terms.

d. Calculation of Continuous System Transition Matrix

If the continuous system matrix $\underline{A}_{cc}$ has γ independent eigenvectors, the exponential function $e^{\underline{A}_{cc}\tau}$ may be expressed as:

$$e^{\underline{A}_{cc}\tau} = \underline{W}\, e^{\underline{\Lambda}\tau}\, \underline{W}^{-1} \tag{12}$$

where: $\underline{W}$ modal matrix of $\underline{A}_{cc}$ eigenvectors
$\underline{\Lambda}$ diagonal matrix of $\underline{A}_{cc}$ eigenvalues

The second term in the $\underline{\Phi}(\tau)$ matrix may be expressed as:

$$\underline{A}_{cc}^{-1}\left[e^{\underline{A}_{cc}\tau} - \underline{I}\right]\underline{A}_{cs} = \underline{W}\,\underline{\Lambda}^{-1}\left[e^{\underline{\Lambda}\tau} - \underline{I}\right]\underline{W}^{-1}\underline{A}_{cs} \tag{13}$$

This spectural factorization approach is used by the EASY program to calculate the transition matrix Ψ .

If the continuous system stability matrix $\underline{A}_{cc}$ does not have Υ independent eigenvectors, this is detected by the program and a Pade approximation method is used to calculate $e^{\underline{A}_{cc}\tau}$. This occurs in a continuous system in which components with exactly the same eigenvalues appear in a series connection. The sixth order Pade approximation is:

$$e^{\underline{A}_{cc}\tau} = \left[\underline{I} - \frac{\tau}{2}\underline{A}_{cc} + \frac{\tau^2}{10}\underline{A}_{cc}^{\,2} - \frac{\tau^3}{120}\underline{A}_{cc}^{\,3}\right]^{-1}\left[\underline{I} + \frac{\tau}{2}\underline{A}_{cc} + \frac{\tau^2}{10}\underline{A}_{cc}^{\,2} + \frac{\tau^3}{120}\underline{A}_{cc}^{\,3}\right] \tag{14}$$

3. Combined System Transition Matrix

It is proved by Kalman and Bertram in reference 1 that the stability of a periodic system is determined by the eigenvalues of the combined system transition matrix, that is, the *transition matrix* that describes one complete system of the system operation.

a. Single Sample Rate

For a single sample rate system, the transition matrix would be obtained by the product of one $\underline{B}$ matrix, as given in (4) with one $\underline{\Phi}$ matrix as given in (10). Such a system is shown in Figure 17. The continuous system stability matrix for this system would be:

$$A = \begin{bmatrix} -10 & 0 & 10 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{15}$$

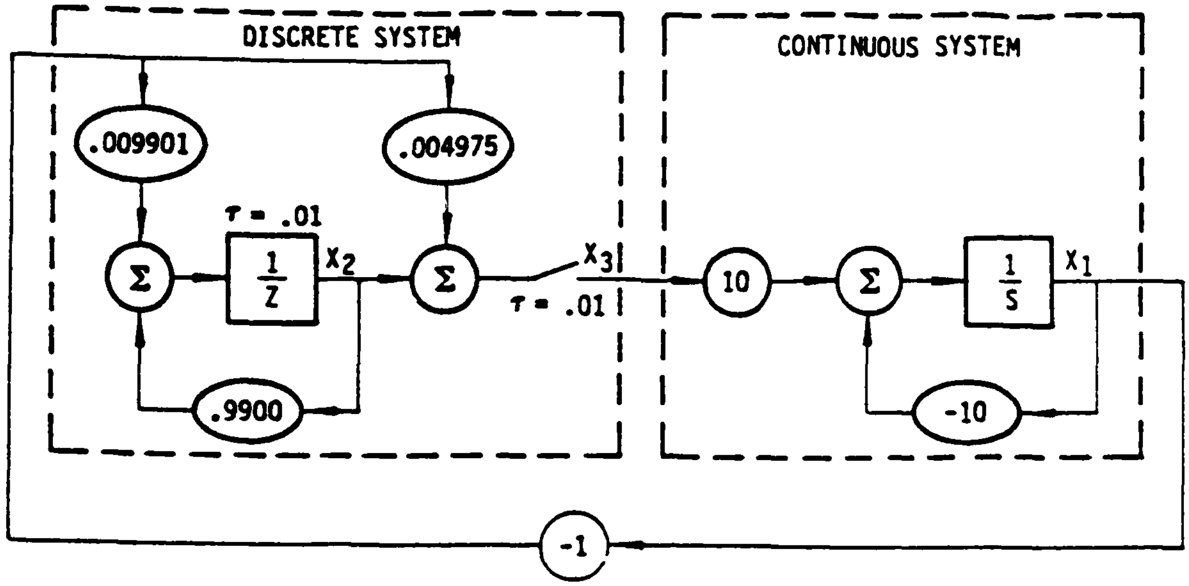


Figure 17. Single Sampling Rate Example

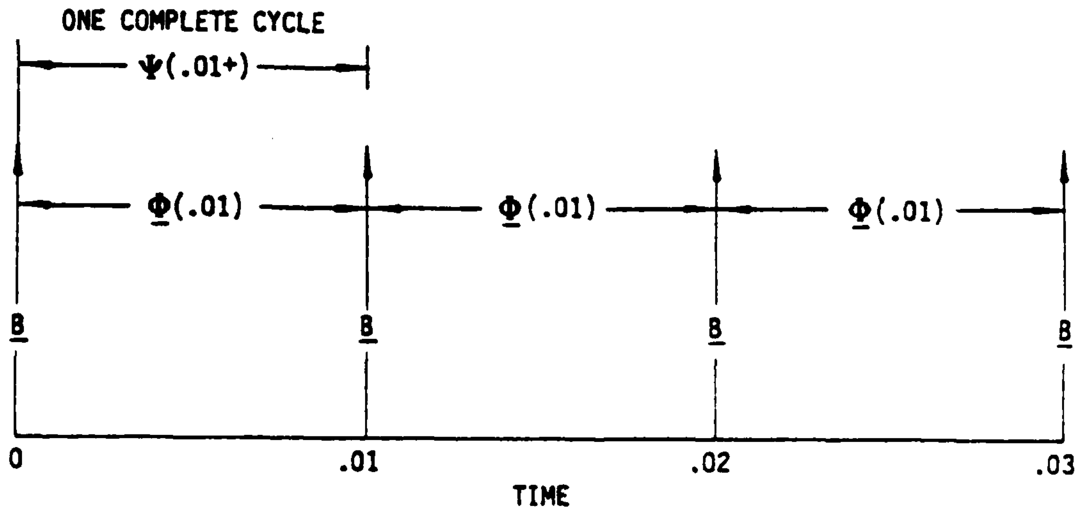


Figure 18. Pictorial Representation of Single Sampling Rate Transition Matrices

The discrete system transition matrix would be:

$$\underline{B} = \begin{bmatrix} 1.0 & 0 & 0 \\ -.009901 & .9900 & 0 \\ -.004975 & 1.0 & 0 \end{bmatrix} \tag{16}$$

The continuous system transition matrix for this system is:

$$\underline{\Phi}(.01) = e^{.01\underline{A}} = \begin{bmatrix} .904837 & 0 & .0951625 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

A complete cycle of this system occurs after one sample period as shown in Figure 18. The system transition is given by:

$$\underline{X}(.01+) = \underline{\Phi}(.01)\,\underline{B}\,\underline{X}(0) = \underline{\Psi}(.01+)\,\underline{X}(0) \tag{18}$$

The total system transition matrix:

$$\underline{\Psi}(.01+) = \begin{bmatrix} .90436 & .09516 & 0 \\ -.009901 & .9900 & 0 \\ -.004975 & 1. & 0 \end{bmatrix} \tag{19}$$

$$\underline{\Psi}(.01+) = \begin{bmatrix} .90436 & .09516 \\ -.009901 & .9901 \end{bmatrix} \tag{20}$$

Note that the final system transition matrix product is shown as a 2 x 2 rather than a 3 x 3 matrix. The sample state, X_3, has a zero column in the final transition matrix, and therefore, contributes nothing to the state of the system at the next sample period. The row and column corresponding to this state may, therefore, be dropped from the total system stability matrix at this point in the analysis. This will occur in general for all sample states in a model. However, in order to express the total system transition matrix as a simple product of matrices, it is necessary to carry the sample states along in the matrix calculation until the final transition matrix is formed.

b. Integer Multiple Sampling Rate

For a multiple sampling rate system, we will first consider the special case where the larger sample periods are all integer multiples of all smaller sample periods. An example of such a system is shown in Figure 19. Here the sampling periods are: $\tau_1 = .01$ and $\tau_2 = .04$. Our objective is to build the total system transition matrix, $\underline{\Psi}(.04+)$, that spans one complete cycle of the multirate system as shown in Figure 20, one complete cycle occurs for this system after four samples of the fastest sampling rate. The total system transition matrix, $\underline{\Psi}(.04+)$, can be expressed as:

$$\underline{\Psi}(.04+) = \left[\underline{\Phi}(.01)\,\underline{B}_{.01}\right]^4 \underline{B}_{.04} \tag{22}$$

by means of the transition property of transition matrices. For the multirate case, there is a $\underline{B}$ matrix for each sampling rate. The multirate $\underline{B}$ matrices shown in (22) differ only slightly from the single rate form of (5). They are of the form:

$$\underline{B} = \begin{bmatrix} \underline{I} & \underline{0} & \underline{0} \\ \underline{B}_{dc} & \underline{B}_{dd} & \underline{0} \\ \underline{B}_{sc} & \underline{B}_{sd} & \underline{B}_{ss} \end{bmatrix} \tag{23}$$

The rows of $\underline{B}_\tau$ corresponding to discrete states which do not change at period τ are equal to the corresponding row from an identity matrix. The

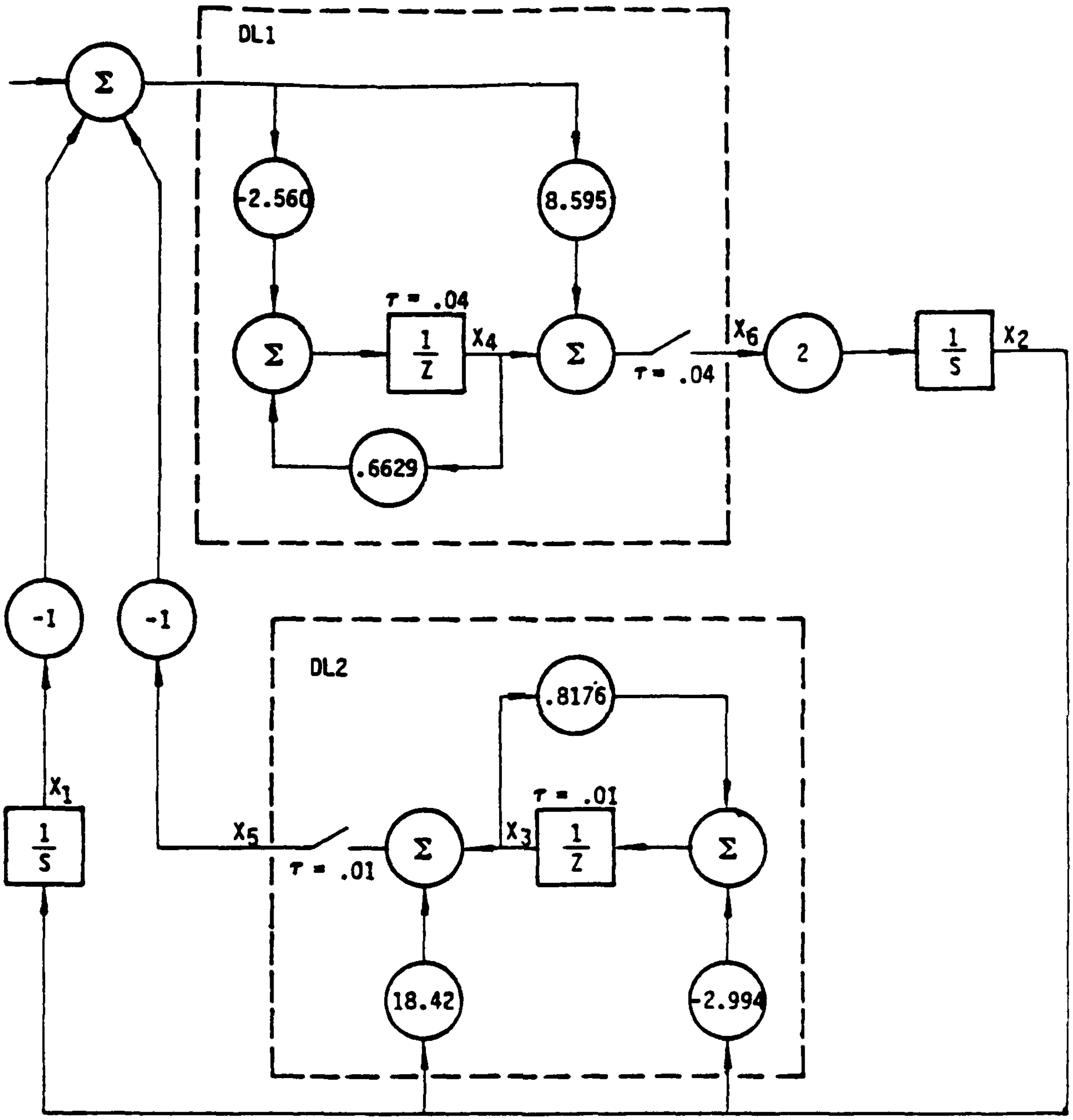


Figure 19. Multisampling Rate Example

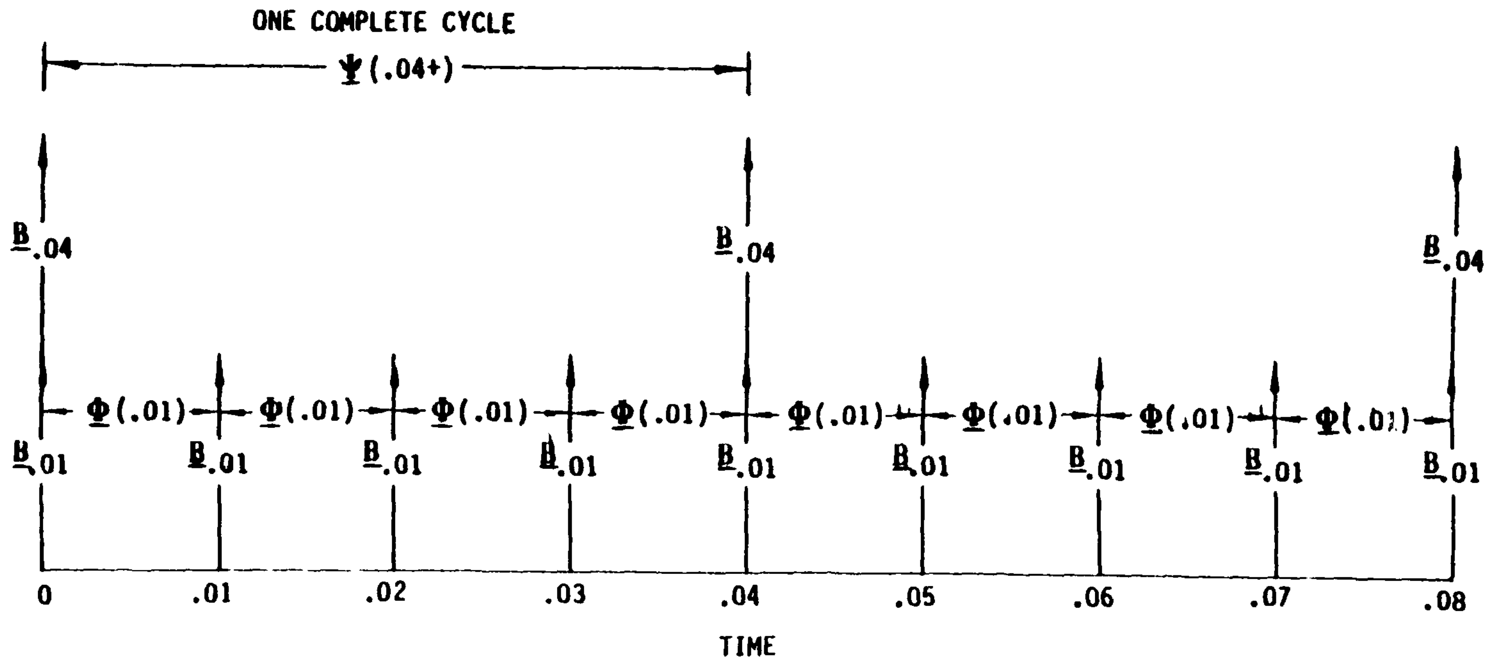
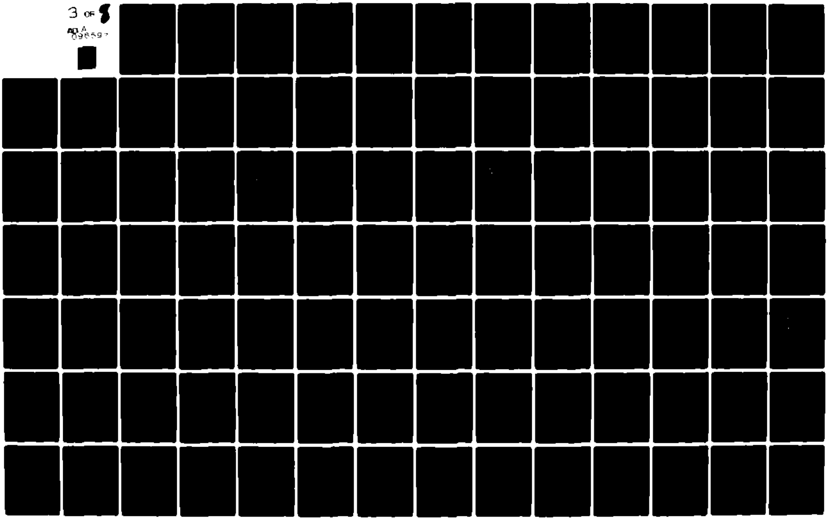


Figure 20. Pictorial Representation of Multisampling Rate Transition Matrices-Integer Multiple Rates

AD-A096 597 BOEING MILITARY AIRPLANE CO SEATTLE WA F/G 1/3
ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM. VOLUME --ETC(U)
SEP 80 C L WEST, B R UMMEL, R F YURCZYK F33615-79-C-3407
UNCLASSIFIED AFWAL-TR-80-3014-VOL-1 NL

3 OF 5
AD A
096597



rows of $\underline{B}_{\tau}$ corresponding to sampler states of the period τ have zero elements in $\underline{B}_{ss}$. Thus the only difference between $\underline{B}_{\tau}$ and the $\underline{B}$ matrix shown in (5) is the possible addition of ones on the diagonal of $\underline{B}_{ss}$ for those sample states corresponding to periods other than τ.

This may be seen by examining the matrices for the example system of Figure 19.

Continuous system stability matrix:

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{24}$$

Discrete transition matrix for sample period .01:

$$\underline{B}_{.01} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -2.994 & .8176 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 18.42 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{25}$$

Discrete transition matrix for sample period .04:

$$\underline{B}_{.04} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 2.560 & 47.15 & 2.560 & .6629 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -8.595 & -158.3 & -8.595 & 1 & 0 & 0 \end{bmatrix} \tag{26}$$

One point should be made regarding the model of Figure 19. The sample state X_5 is redundant since it is in a path that only leads to other discrete states. Sample states are normally used only in paths that lead from delay states to continuous states. In order to simplify the assembly of discrete system models, the EASY program models of all digital filters

contain a sample state on their output. However, during the calculation of the $\underline{B}$ matrices by the EASY Analysis program, these samplers are treated as being closed, for all sample periods which are modulo their sample rate. This causes the sampler X_5 to pass information from continuous state X_2 and delay state X_3 on to discrete states X_4 and X_6. Thus, the $\underline{B}_{.04}$ matrix has the correct no-zero elements (4,2), (4,3), (6,2), and (6,3) that would occur if the sample state X_5 had been omitted from the model.

The functional form of equation (22) can be extended to any number of sampling rates as long as each larger sample period is an integer multiple of the next lower sample period. Thus if:

$$\begin{aligned} N_2 &= \tau_2/\tau_1 \\ N_3 &= \tau_3/\tau_2 \\ &\vdots \\ N_n &= \tau_n/\tau_{n-1} \end{aligned} \tag{27}$$

then the total system transition matrix is

$$\underline{\Psi} = \Big\{ \cdot \cdot \cdot \Big[\left(\underline{\Phi}\, \underline{B}_{\tau_1} \right)^{N_2} \underline{B}_{\tau_2} \Big]^{N_3} \underline{B}_{\tau_3} \cdot \cdot \cdot \cdot \Big\}^{N_n} \underline{B}_{\tau_n} \tag{28}$$

The EASY program is currently dimensioned for n = 10, i.e., up to ten different sampling rates may occur in one model.

c. Noninteger Multiple Sampling Rates

For noninteger multiple sampling rates, the simple expression of (28) cannot be used. However, the same technique of building up the total system transition matrix from a continuous system transition matrix and a series of discrete system transition matrices still applies. For example, consider the system shown in Figure 19 with sample periods of 0.02 and 0.03 in place of 0.01 and 0.04. Figure 21 shows a pictorial representation of the transitions that take place to complete a cycle.

The total system transition matrix can be expressed in terms of the basic transition matrices as follows:

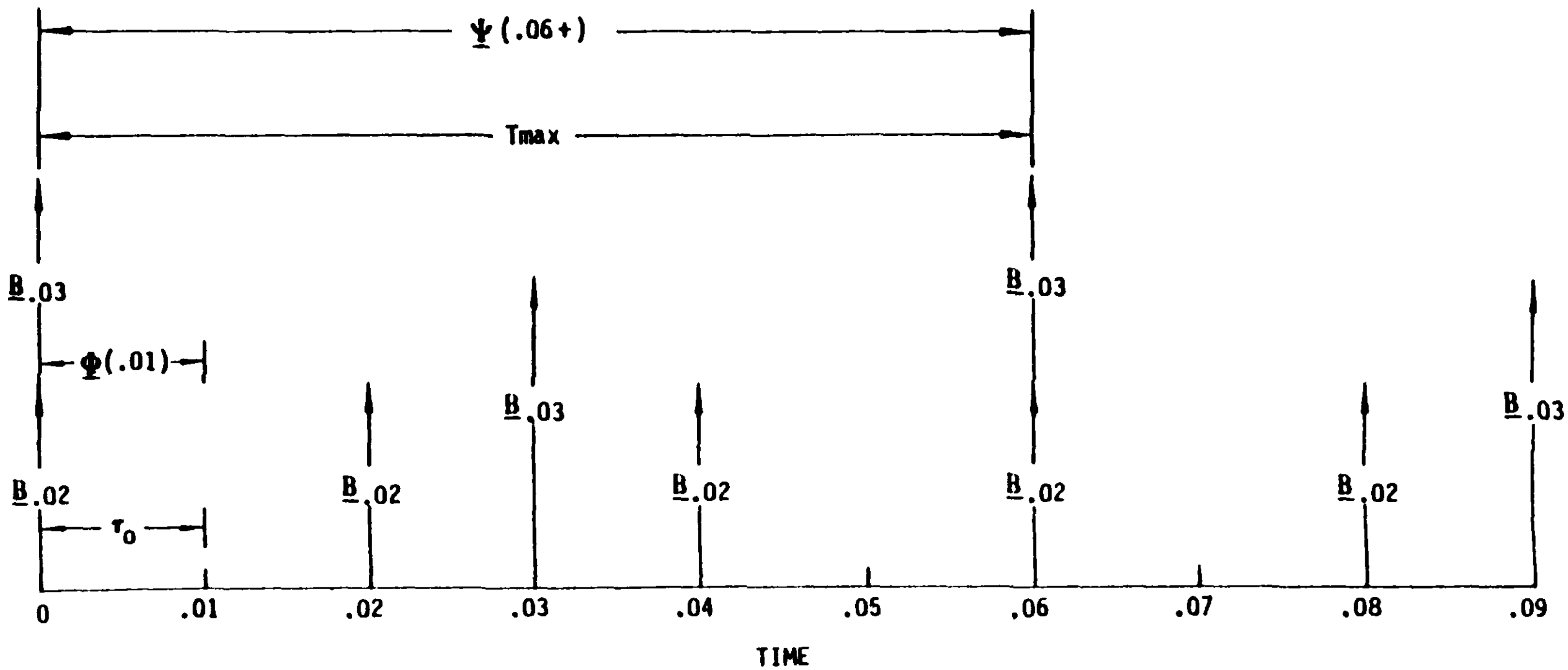


Figure 21. Pictorial Representation of Multisampling Rate Transition Matrices-Noninteger Multiple Rates

$$\underline{\Psi}(.06+) = \underline{\Phi}^2(.01)\,\underline{B}_{.02}\,\underline{\Phi}(.01)\,\underline{B}_{.03}\,\underline{\Phi}(.01)\,\underline{B}_{.02}\,\underline{\Phi}^2(.01)\,\underline{B}_{.02}\,\underline{B}_{.03} \tag{29}$$

In this case it is necessary to introduce a continuous system transition matrix that spans a period, .01, which is less than the smallest given sampling period, $\tau_1 = .02$. The total system period, .06, is also greater than the largest given sampling period, $\tau_2 = .03$.

In general, the continuous system transition matrix is required for a period, τ_0, which is the greatestcommon divisor of the sample periods:

$$\tau_0 = \text{g.c.d.}\ (\tau_1, \tau_2, \ldots, \tau_n) \tag{30}$$

The total system period, T_{max}, will be the least common multiple of the sample periods.

$$T_{max} = \text{l.c.m.}\ (\tau_1, \tau_2, \ldots, \tau_n) \tag{31}$$

In order to form the total system transition matrix, the quantities τ_0 and T_{max} are calculated. The total period T_{max} is then scanned in increments of τ_0 and the appropriate power of $\underline{\Phi}(\tau_0)$, and $\underline{B}_\tau$ matrices are multiplied together to form the total system transition matrix. This capability is not currently available in the EASY program.

REFERENCES

1. Burroughs, J. D., et al, "Environmental Control System (ECS) Transient Analysis", AFFDL-TR-77-102, The Boeing Company, October 1977.

2. McCarty, R. E., "Simulation of the Dynamic Tensile Characteristics of Nylon Parachute Materials", AFFDL-TR-78-169, November 1978.

APPENDIX A

EASY5 - MODEL GENERATION - COMMANDS

Format	Description
ADD PARAMETERS = $q_1,q_2(n_1,n_2)$	Add parameters to model (also dimensions)
ADD TABLES = $t_1,n_1,n,t_2,n_2,n,$	Add tables to model
ADD VARIABLES = $q_1,q_2(n_1,n_2)$	Add variables to model (also dimensions)
*Comment #	Add comment to model description
DEBUG	Add debug print statements to model
DIAGNOSTIC CONTROL = n	Control diagnostic printout from model generation program
END OF MODEL	Specify end of model description
FORT	Specify user Fortran Component
FORTRAN STATEMENTS L_1 L_2 . . .	Specify start of FORTRAN statements
INPUTS = $C_1(q_{out} = q_{in})$, FORT$(q_{out} = q_{in})$	Specify source of inputs to components
LIST STANDARD COMPONENTS	Request listing of standard components
LOCATION = n_1 n_2 n_3	Specify component location on schematic
Matrix arithmetic #	Compact Matrix Algebra
MODEL DESCRIPTION = test	Specify start of model description
O.C. ANALYSIS	Specify only analyses-no O.C. DESIGN
O.C. CRITERIA = $q_1,q_2,\ldots$	Specify O.C. criteria variables
O.C. INPUTS = q_1,q_2	Specify O.C. input variables
O.C. MODEL ORDER = n	Specify model order to be used for O.C. DESIGN
O.C. ORDER = n	Specify optimal controller order
O.C. OUTPUTS = $q_1,q_2,\ldots$	Specify O.C. output variables

Format	Description
PRINT	Request printed model output
Standard Components #	Standard Components----see list
C,N = n_1, M = n_j	Dimension Standard Component
TABLE DIMENSION = t_1=n_1, t_2=n_2,...	Specify table standard component Table dimensions
/*EOR #	End of record for mini-time-share file

#Not a command

Modifier Notations	Phrase Delimiters
C_1 - Standard component name	= equal sign
L_1 - Line of FORTRAN source code	, comma
n_1 - Integer number	(left parenthesis
q_1 - Input or output quantity name	) right parenthesis
t_i - Table name	three or more blanks

APPENDIX B

EASY5 - ANALYSIS - COMMANDS

ALL STATES	Activate all model states (DEFAULT)
CALCOMP	Requests plots on CalComp plotter
CALC XIC	Allows manual I.C. calculations
DEFINE PARAMETERS = p_1=p_2,...	Define parameter names
DEFINE RATES = r_1=r_2,...	Define rate names
DEFINE STATES = s_1=s_2,...	Define state names
DEFINE VARIABLES = v_1=v_2,...	Define variable names
DESIGN O.C.	Initiate optimal controller design
DISPLAYi i = 1,2,3,4,5,6	Specify quantities to be plotted
q_1,vs,TIME	(5 plots/display 6 displays = max 30 plots
Max	3000 points/display set)
q_2,vs,q_3	
. . .	
EIGEN SENSITIVITY	Initiate eigenvalue sensitivity calculation
EIGEN PARAMETER = p_i	
ERROR CONTROL = s_1=n_1,...	Specify integrator error controls
INITIAL CONDITIONS = s_1=n_1,...	Specify initial conditions/operating point
INITIAL TIME = n	Specify initial value of time
INT CONTROL = s_1=n_1,...	Activate or freeze model states
LINEAR ANALYSIS	Initiate linear analysis
Matrix Parameters*	Input matrix parameter values
MTS PLOTS	Requests plots on MTS plotter
NO STATES	Freeze all model states
O.C. DATA	Input optimal controller data
YOP;UOP;Q;RU;CD;	
CS;G;S;A;FK	

OMIT PLOT POINTS	Omit boxes around plot points
OMIT TABLE PRINTOUT	Omit print back of table inputs
PARAMETER VALUES = p_1=n_1,...	Input parameter values
PLOT ALL TABLES	Request plots of <u>all</u> tables
PLOT ID = text	Specify plot identification
PLOT OFF	Deactivate plotting (DEFAULT)
PLOT ON	Activate plotting
PLOT TABLES = t_1,t_2,...	Requests plots of specified tables
PRINT	Initiate single pring pass through model
PRINT2	Specify second print option
PRINT VARIABLES = q_1,...q_{10}	Specify columnar option print variables (PRINT CONTROL=5)
PRINTER PLOTS	Requests plots on line printer
ROOT LOCUS	Initiate root locus analysis
RL PARAMETER = p	Specify root locus parameter
RL START = n	Specify initial value of RL PARAMETER
RL STOP = n	Specify final value of RL PARAMETER
RL POINTS = n	Specify number of root locus points
RL MANUAL SCALES	Request manual root locus plot scales
REAL MIN = n	Real axis minimum scale value
REAL MAX = n	Real axis maximum scale value
IMAG MIN = n	Imaginary axis min. scale value
IMAG MAX = n	Imaginary axis max. scale value
RL AUTO SCALES	Request auto plot scales (DEFAULT)
SAVE O.C.	Write optimal controller arrays to TAPE3
SCAN1	Initiate one dimensional function scan
DEPEN = q	Specify 2nd dependent variable
START2 = n	Specify initial value of INDEP2
DELTA2 = n	Specify increment size for INDEP2
CURVES2 = n	Specify number of values for INDEP2
(Also requires DEPEN, INDEP1, START1, STOP1)	

SC4020	Request plots on SC4020 microfilm
SIMULATE	Initiate simulation (Time History)
PRINT CONTROL = n	Specify print option
PRINT2 = n	
PRATE = n	Request printout every n plot intervals
PRATE2 = n	
OUTRATE = n	Request plot points every n*TINC
OUTRATE2 = n	
INT MODE = n	Specify integration method
TINC = n	Specify integrator report interval
TINC2 = n	
TMAX = n	Specify time history duration
SI MANUAL SCALES	Request manual simulation plot scales
SI AUTO SCALES	Request auto plot scales (DEFAULT)
STABILITY MARGINS	Initiate stability margin calculation
SM PARAMETERS = $p_1,\ldots p_{10}$	Specify stability margin parameters
STEADY STATE	Initiate steady state calculation
SS PARAMETER = p	Specify SS parameter (optional)
SS START = n	Specify initial value of SS PARAMETER
SS STOP = n	Specify final value of SS PARAMETER
SS POINTS = n	Specify number of SS calculations
SS ITERATIONS = n	Specify number if iterations to be used
SS MANUAL SCALES	Request manual plot scales
SS AUTO SCALES	Request auto plot scales (DEFAULT)
TABLE = t ,n ,n ,n (table data)	Input tabular data
TITLE = text	Specify plot title

TRANSFER FUNCTION	Initiate transfer function calculation
TF INPUT = q	Specify transfer function input quantity
TF OUTPUT = q	Specify transfer function output quantity
BODE	Request Bode format for plots
NICHOLS	Request Nichols format for plots
NYQUIST	Request Nyquist format for plots
TF MANUAL SCALES	Request manual plot scales
FREQ MIN = n	Specify minimum frequency r.p.s.
FREQ MAX = n	Specify maximum frequency r.p.s.
TF AUTO SCALES	Request auto plot scales (DEFAULT)
XIC-X	Transfer state to initial condition vector
XIC_i-XIC i=1,2,3	Transfer XIC to one of 3 storage vectors
XIC-XIC_i i=1,2,3	Retrieve XIC from one of 3 storage vectors
/*EOF	End of file for mini-time-share file

#Not a Command

Modifier Notations

n_i - numeric value

p_i parameter name

q_i - parameter, variable, state, or rate name

r_1 - rate name

s_i - state name

t_i - table name

v_i - variable name

Phrase Delimiters

= equal sign

, comma

(left parenthesis

(right parenthesis

three or more blanks

APPENDIX C

ANALYSIS CHECKLISTS

Before requesting any of the EASY5 analyses, certain program commands should be issued to assure that the analysis will be successful. These program commands will place the system model in the proper configuration and complete the analysis specification. The following pages provide check lists of program commands that snould be considered before requesting each analysis. The analyses are listed in alphabetical order.

LINEAR ANALYSIS

Model Data

- TITLE
- PARAMETER VALUES
- TABLES
- INITIAL CONDITIONS

Integrator Configuration

- INT CONTROL
- ERROR CONTROL

O.C. DESIGN

Model Data

- TITLE
- PARAMETER VALUES
- TABLES
- INITIAL CONDITIONS
- O.C. DATA: YOP,UOP,Q,RU,CD,CS

Integrator Configuration

- ALL STATES
- ERROR CONTROL

ROOT LOCUS

Model Data

- TITLE
- PARAMETER VALUES
- TABLES
- INITIAL CONDITIONS

Integration Configurations

- INT CONTROL
- ERROR CONTROL

Root Locus Specifications

- RL PARAMETER
- RL START
- RL STOP
- RL POINTS

Output Controls

- RL MANUAL SCALES
- RL AUTO SCALES
- REAL MIN
- REAL MAX
- IMAG MIN
- IMAG MAX

SCAN1, SCAN2

Model Data

PARAMETER VALUES

- TITLE
- PARAMETER VALUES
- TABLES
- INITIAL CONDITIONS

Scan Specifications

DEPEN
INDEP1
INDEP2
START1
STOP1
START2
DELTA2
CURVES2

	<u>SIMULATE</u>
Integration Control	
TINC	TINC2
TMAX	
INT MODE	
ERROR CONTROL	
INT CONTROL	
Output Controls	
OUTRATE	OUTRATE2
PRATE	PRATE2
PRINT CONTROL	PRINT2
DISPLAY1, 2, 3, 4, 5	
PLOT ON	
PLOT TITLE	
PLOT ID	
SI MANUAL SCALES	
SI AUTO SCALES	
PRINTER PLOTS	
PRINT2 FROM, __, TO, __	

STABILITY MARGINS

Model Data

TITLE

PARAMETER VALUES

TABLES

INITIAL CONDITIONS

Integration Configuration

INT CONTROL

ERROR CONTROL

Stability Margin Specification

SM PARAMETERS

STEADY STATE

Model Data

TITLE

PARAMETER VALUES

TABLES

INITIAL CONDITIONS

Integration Configuration

INT CONTROL

ERROR CONTROL

Note: Steady state cannot be found for system with eigenvalue at origin.

Output Controls

PRINT CONTROL

DISPLAY1, 2, 3, 4, 5, 6

PLOT ON

PRINTER PLOT

PLOT TITLE

PLOT ID

SS MANUAL SCALES

SS AUTO SCALES

Steady State Specifications

SS PARAMETER

SS START

SS STOP

SS POINTS

SS ITERATIONS

TRANSFER FUNCTION

Model Data

TITLE

PARAMETER VALUES

TABLES

INITIAL CONDITIONS

Integrator Configuration

INT CONTROL

ERROR CONTROL

Transfer Function Specification

TF INPUT

TF OUTPUT

BODE, NICHOLS, NYQUIST

Output Controls

TF MANUAL SCALES

TF AUTO SCALES

FREQ MIN

FREQ MAX

APPENDIX D

EASIEST INPUT/OUTPUT LISTS AND ASSOCIATED FIGURES

This appendix contains input and output tables for all the EASIEST standard components. Descriptive figures are also presented for the more complex components.

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
WT			WEIGHT OF THE ATTACHED BODY	LB
BMI(3)			ATTACHED BODY MOMENTS OF INERTIA (IXX,IYY,IZZ)	SLUG-FT^2
BPI(3)			ATTACHED BODY PRODUCTS OF INERTIA (IXY,IXZ,IYZ)	SLUG-FT^2
FAB(3)*		RS	X,Y,Z BODY AXIS FORCE COMPONENTS	LB
TAB(3)*		RS	X,Y,Z BODY AXIS TORQUE COMPONENTS	FT-LB
FAU(3)*			AUXILIARY X,Y,Z BODY AXIS FORCE COMPONENTS	LB
TAU(3)*			AUXILIARY X,Y,Z BODY AXIS TORQUE COMPONENTS	FT-LB
TRM(3)*		RS	X,Y,Z PARENT BODY EARTH VELOCITY COMPONENTS FOR CALCULATING THE LINEAR POSITION RATES DURING TRIM	FT/SEC

*Default value = 0

NAME	PORT NO.	DESCRIPTION	UNITS
UAB(3)*		X,Y,Z BODY AXIS LINEAR VELOCITY VECTOR OF THE ATTACHED BODY	FT/SEC
XAB(3)*		X,Y,Z EARTH LINEAR POSITION VECTOR OF THE ATTACHED BODY	FT
WAB(3)*		X,Y,Z BODY AXIS ANGULAR VELOCITY VECTOR OF THE ATTACHED BODY	DEG/SEC
EAB(3)*		EARTH TO ATTACHED BODY EULER ANGLES (YAW, PITCH, ROLL)	DEG

*These output quantities are states

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
AW			AIRPLANE WEIGHT	LB
B			WINGSPAN OF AIRPLANE	FT
C			MEAN AERODYNAMIC CHORD	FT
S			REFERENCE AREA	FT^2
XCP			AIRPLANE X-AXIS POSITION OF THE CENTER OF PRESSURE	FT
AMI(3)			MOMENT OF INERTIA VECTOR ABOUT THE AIRPLANE C.G. (IXX,IYY,IZZ)	$SLUG\text{-}FT^2$
API(3)			PRODUCT OF INERTIA VECTOR ABOUT THE AIRPLANE C.G. (IXY,IXZ,IYZ)	$SLUG\text{-}FT^2$
THR*			EXTERNAL THRUST SETTING	LB
AIL*			EXTERNAL AILERON SETTING	DEG
ELE*			EXTERNAL ELEVATOR SETTING	DEG
RUD*			EXTERNAL RUDDER SETTING	DEG
XEN(3)			X,Y,Z AIRPLANE BODY AXIS POSITION VECTOR OF THE ENGINE	FT
END(3)			AIRPLANE BODY AXIS DIRECTION COSINES OF THE ENGINE THRUST VECTOR	—
TAL			DESIRED TRIM AIRPLANE ALTITUDE	FT
TVE			DESIRED TRIM AIRPLANE VELOCITY	FT/SEC

*Default value = 0

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
FRA(3)*	1	RL	X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE RAILS	LB
TRA(3)*	1	RL	X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE RAILS	FT-LB
FCA(3)*	1	CT	X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE CATAPULT	LB
TCA(3)*	1	CT	X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE CATAPULT	FT-LB
FDA(3)*	1	DR	X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE DART	LB
TDA(3)*	1	DR	X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE DART	FT-LB
FRA(3)*	2	RL	X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE RAILS	LB
TRA(3)*	2	RL	X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE RAILS	FT-LB
FCA(3)*	2	CT	X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE CATAPULT	LB

*Default value = 0.

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
TCA(3)*	2	CT	X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE CATAPULT	FT-LB
FDA(3)*	2	DR	X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS ACTING ON THE AIRPLANE FROM THE DART	LB
TDA(3)*	2	DR	X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS ACTING ON THE AIRPLANE FROM THE DART	FT-LB
CPF			PRINT FLAG FOR THE AERO-DYNAMIC COEFFICIENTS	—

*Default value = 0

NAME	PORT NO.	DESCRIPTION	UNITS
UAP(3)*		X,Y,Z AIRPLANE BODY AXIS LINEAR VELOCITY VECTOR OF THE AIRPLANE	FT/SEC
XAP(3)*		X,Y,Z EARTH LINEAR POSITION VECTOR OF THE AIRPLANE	FT
WAP(3)*		X,Y,Z AIRPLANE BODY AXIS ANGULAR VELOCITY VECTOR OF THE AIRPLANE	DEG/SEC
EAP(3)*		EARTH TO AIRPLANE EULER ANGLES (YAW, PITCH, ROLL)	DEG
TRM(4)*		TRIM CONTROL SETTINGS 1) THROTTLE 2) AILERON 3) ELEVATOR 4) RUDDER	–
ALP		AIRPLANE ANGLE OF ATTACK	DEG
BET		AIRPLANE SIDESLIP ANGLE	DEG
VM		AIRPLANE MACH NUMBER	–
ALT		AIRPLANE ALTITUDE ABOVE SEA LEVEL	FT

*These output quantities are states

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
H			REFERENCE ALTITUDE WITH RESPECT TO SEA LEVEL	FT
WIN(3)			X,Y,Z INERTIAL SYSTEM WIND COMPONENTS	FT/SEC
BP*			BAROMETRIC PRESSURE AT REFERENCE ALTUTIDE	IN HG
TE			TEMPERATURE AT REFERENCE ALTITUDE	DEG F
SW**			GRAVITY SWITCH FOR UNSUPPORTED SEAT 0 = GRAVITY OFF 1 = GRAVITY ON	-

NAME	PORT NO.	DESCRIPTION	UNITS
VS		VELOCITY OF SOUND	FT/SEC
RHO		AIR DENSITY	$SLUG/FT^3$

*Default value = 0
**Default value = 1

NOTE: H, BP, AND TE MUST BE INITIALIZED FOR A NON-STANDARD ATMOSPHERE. A STANDARD ATMOSPHERE IS ESTABLISHED WHEN BP EQUALS ZERO (DEFAULT)

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
FL			FLAG TO INITIATE AEROMED CALCULATIONS (1 = START)	–
PRT			FLAG SENT TO PROGRAM AEROMED TO PRINT THE AEROMEDICAL VARIABLES (1 = PRINT) **DEFAULT = 0**	–
EXP			MEDICAL INJURY EXPONENT **DEFAULT = 2**	–
GXP			THE LIMIT VALUE FOR THE X-AXIS POSITIVE AEROMED LOAD FACTOR **DEFAULT = 35**	G's
GXN			THE LIMIT VALUE FOR THE X-AXIS NEGATIVE AEROMED LOAD FACTOR **DEFAULT = 30**	G's
GYL			THE LIMIT VALUE FOR THE Y-AXIS AEROMED LOAD FACTOR **DEFAULT = 15**	G's
GZL			THE LIMIT VALUE FOR THE Z-AXIS NEGATIVE AEROMED LOAD FACTOR **DEFAULT = 12**	G's
DRP			LIMIT VALUE OF THE DYNAMIC RESPONSE WHEN THE ACCELERATION VECTOR IS FORWARD OF THE PLANE OF THE SEAT BACK **DEFAULT = 18**	–
DRN			LIMIT VALUE OF THE DYNAMIC RESPONSE WHEN THE ACCELERATION VECTOR IS AFT OF THE PLANE OF THE SEAT BACK **DEFAULT = 16**	–

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
RDL			ACCELERATION RADICAL LIMIT	—
DR		SE or CE	DYNAMIC RESPONSE	—
GX		SE or CE	X-AXIS LOAD FACTOR	G's
GY		SE or CE	Y-AXIS LOAD FACTOR	G's
GZ		SE or CE	Z-AXIS LOAD FACTOR	G's

NAME	PORT NO.	DESCRIPTION	UNITS
DRE		DYNAMIC RESPONSE	–
RAD		ACCELERATION RADICAL	–
PTS		CURRENT NUMBER OF DATA SETS WRITTEN TO TAPE 7	–
PTI		VALUE OF TIME WHEN THE LAST DATA SET WAS WRITTEN TO TAPE 7	SEC

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
TCX			PLATE SYSTEM X-AXIS FORCE COEFFICIENT TABLE:	
			PLATE ANGLE OF ATTACK (INDEPENDENT)	DEG
			PLATE X-AXIS COEFFICIENT (DEPENDENT)	—
TCZ			PLATE SYSTEM Z-AXIS FORCE COEFFICIENT TABLE:	
			PLATE ANGLE OF ATTACK (INDENPENDENT)	DEG
			PLATE Z-AXIS COEFFICIENT (DEPENDENT)	
UP**			EJECTION DIRECTION FLAG WITH RESPECT TO THE AIRPLANE 1 = UPWARD -1 = DOWNWARD	—
XPC(3)			X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE PLATE CENTROID	FT
PA			REFERENCE AREA OF THE ATTACHED PLATE	FT^2
EPL(3)			SEAT TO PLATE EULER ANGLES	DEG
ZEM*			AIRPLANE BODY Z-AXIS POSITION OF THE PLATE CENTROID WHEN IT ENTERS THE WINDSTREAM	FT
SRP(3)		SE	X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT	FT
UST(3)		SE	X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT	FT/SEC

*Default value = 0.
**Default value = 1.

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
EST(3)		SE	EARTH TO SEAT EULER ANGLES (YAW, PITCH, ROLL)	DEG
WST(3)		SE	X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT	DEG/SEC
XAP(3)*		AE or SL	X,Y,Z EARTH SYSTEM POSITION VECTOR OF THE AIRPLANE CENTER OF GRAVITY	FT
EAP(3)*		AE or SL	EARTH TO AIRPLANE EULER ANGLES	DEG

*Default value = 0

NAME	PORT NO.	DESCRIPTION	UNITS
F2(3)	1	X,Y,Z SEAT BODY AXIS FORCE COMPONENTS OF THE ATTACHED PLATE	LB
T2(3)	1	X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS OF THE ATTACHED PLATE	FT-LB
SW		FLAG SET WHEN THE PLATE CENTROID PENETRATES THE WINDSTREAM (1 = PENETRATION)	—
ALP		PLATE ANGLE OF ATTACK	DEG
CX		X-AXIS FORCE COEFFICIENT	—
CZ		Z-AXIS FORCE COEFFICIENT	—

- STANDARD COMPONENT "AP" CALCULATES THE FORCES AND TORQUES THAT ACT ON THE SEAT FROM AN AERODYNAMIC PLATE
- THE PLATE BODY AXIS FORCE COEFFICIENT TABLES ARE A FUNCTION OF ANGLE OF ATTACK

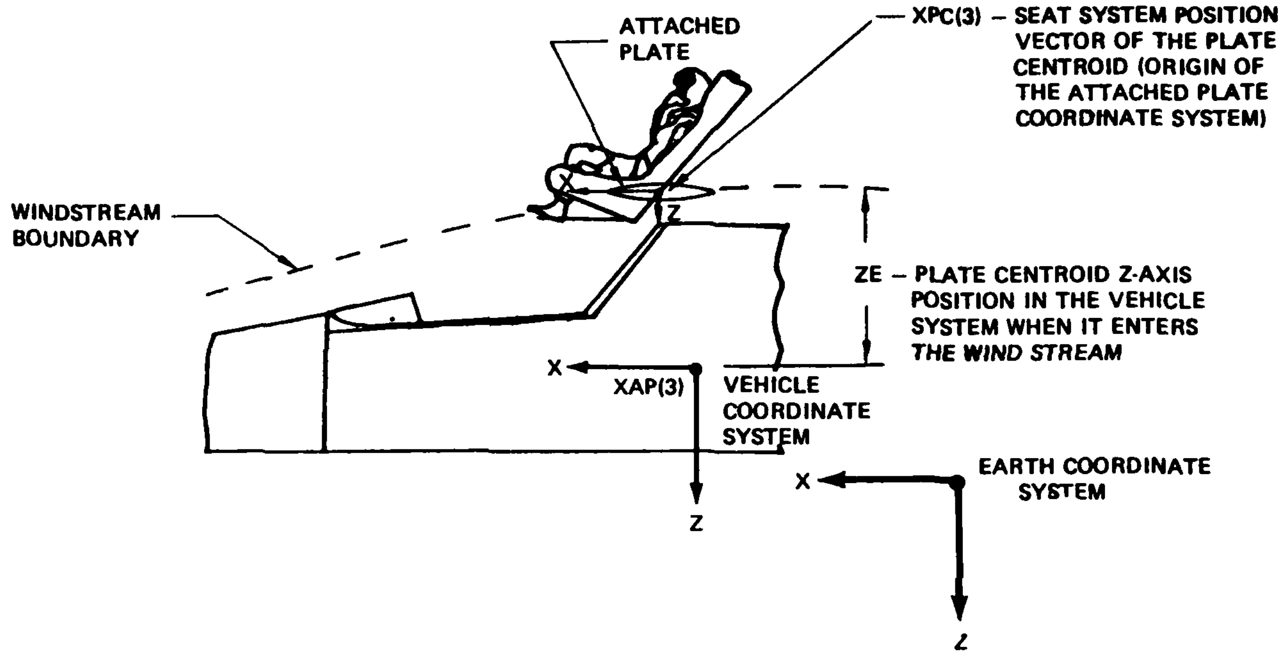


Figure 22. Standard Component "AP" Input/Output Overview

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	AS UNITS
TAE			EXPOSED AREA TABLE: EXPOSED LENGTH (INDEPENDENT) EXPOSED AREA (DEPENDENT)	 FT FT^2
OFF**		RL	FLAG/TO INDICATE SEAT/RAIL SEPARATION (1 = SEPARATION)	—
UP**			EJECTION DIRECTION FLAG WITH RESPECT TO THE AIRPLANE +1 = UPWARD -1 = DOWNWARD	—
ZWS*			AIRPLANE BODY Z-AXIS POSITION OF THE WINDSTREAM BOUNDARY LAYER AT THE POINT OF SEAT PENETRATION	FT
XEM(3)*			X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE POINT ON THE SEAT TO INITIALLY PENETRATE THE WINDSTREAM	FT
CDX*			APPROXIMATE SEAT BODY X-AXIS POSITION OF THE CENTER OF PRESSURE DURING EMERGENCE	FT
ECX**			SEAT BODY X-AXIS EMERGENCE COEFFICIENT	—
ECY**			SEAT Y-AXIS EMERGENCE COEFFICIENT	—
ECZ**			SEAT Z-AXIS EMERGENCE COEFFICIENT	—
CLP*			ROLL DAMPING DERIVATIVE	1/DEG
CMQ*			PITCH DAMPING DERIVATIVE	1/DEG
CNR*			YAW DAMPING DERIVATIVE	1/DEG
S			SEAT REFERENCE AREA	FT^2

*Default value = 0
**Default value = 1

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
SRP(3)		SE	X,Y,Z EARTH LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT	FT
UST(3)		SE	X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT	FT/SEC
EST(3)		SE	EARTH TO SEAT EULER ANGLES (YAW, PITCH, ROLL)	DEG
WST(3)		SE	X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT	DEG/SEC
DSA(3,3)*		RL	SEAT TO AIRPLANE DIRECTION COSINE MATRIX	—
SRA(3)*		RL	X,Y,Z AIRPLANE BODY AXIS LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT	FT
RON*		SR	SUSTAINER ROCKET FLAG (1=ON 0=OFF)	—

*Default = 0

NAME	PORT NO.	DESCRIPTION	UNITS
F2(3)	1	X,Y,Z SEAT BODY AXIS AERODYNAMIC FORCE COMPONENTS	LB
T2(3)	1	X,Y,Z SEAT BODY AXIS AERODYNAMIC TORQUE COMPONENTS	FT-LB
ALP		SEAT ANGLE OF ATTACK	DEG
BET		SEAT SIDESLIP ANGLE	DEG
VM		SEAT MACH NUMBER	—
Q		DYNAMIC PRESSURE	LB
CX		SEAT BODY X-AXIS FORCE COEFFICIENT	—
CY		SEAT BODY Y-AXIS FORCE COEFFICIENT	—
CZ		SEAT BODY Z-AXIS FORCE COEFFICIENT	—
CL		SEAT BODY AXIS ROLLING MOMENT COEFFICIENT	—
CM		SEAT BODY AXIS PITCHING MOMENT COEFFICIENT	—
CN		SEAT BODY AXIS YAWING MOMENT COEFFICIENT	—
EXL		SEAT EXPOSED LENGTH DURING EMERGENCE	FT
EXA		SEAT EXPOSED AREA DURING EMERGENCE	FT^2
CEN(3)		X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE EMERGED AREA CENTROID	FT
TCZ(20)		SEAT Z-AXIS EXPOSED AREA CENTROID LOCATION ARRAY	FT
HD		HYDRAULIC DIAMETER	FT

- STANDARD COMPONENT "AS" CALCULATES THE AERODYNAMIC FORCES AND TORQUES THAT ACT ON THE SEAT

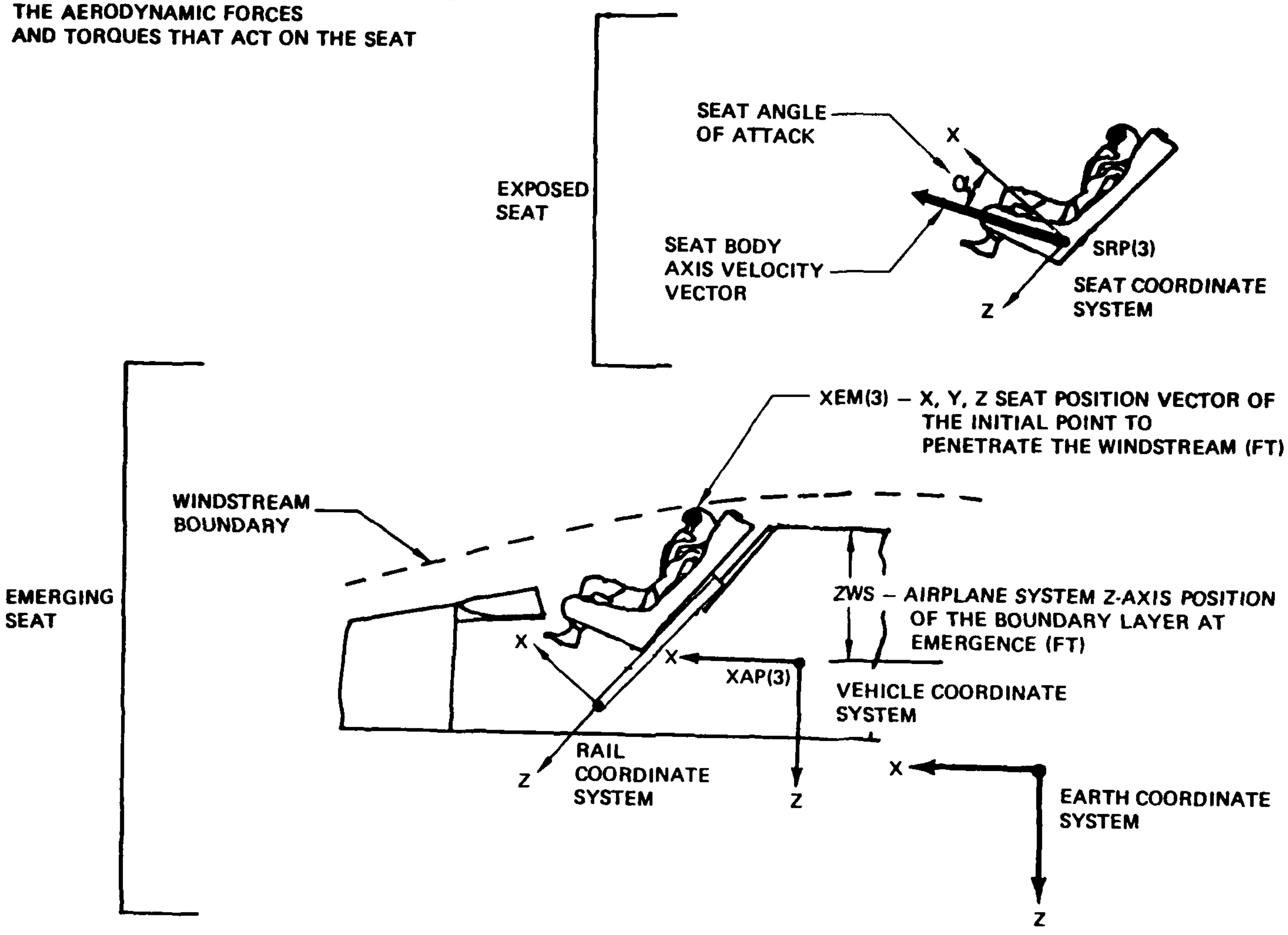


Figure 23. Standard Component "AS" Input/Output Overview

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
SW*			FLAG FOR SEAT/CREWPERSON SEPARATION (1 = SEPARATION)	—
PC			CREWPERSON PERCENTILE	—
CEW			WEIGHT OF THE CREWPERSON CLOTHING AND EQUIPMENT	LB
CMI(3)			CREWPERSON MOMENT OF INERTIA VECTOR ABOUT HIS C.G. (IXX,IYY,IZZ)	SLUG-FT^2
CPI(3)			CREWPERSON PRODUCT OF INERTIA VECTOR ABOUT HIS C.G. (IXY,IXZ,IYZ)	SLUG-FT^2
CLP			AERODYNAMIC ROLL DAMPING COEFFICIENT	1/DEG
CMQ			AERODYNAMIC PITCH DAMPING COEFFICIENT	1/DEG
CNR			AERODYNAMIC YAW DAMPING COEFFICIENT	1/DEG
XSP(3)*			X,Y,Z CREWPERSON SYSTEM POSITION VECTOR OF THE BASE OF THE SPINE	FT
FAB(3)*		RS	X,Y,Z BODY AXIS FORCE COMPONENTS	LB
TAB(3)*		RS	X,Y,Z BODY AXIS TORQUE COMPONENTS	FT-LB
FDO(3)*		LI	X,Y,Z BODY AXIS FORCE COMPONENTS	LB
TDO(3)*		LI	X,Y,Z BODY AXIS TORQUE COMPONENTS	FT-LB

*Default = 0

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
FAU(3)*			X,Y,Z BODY AXIS FORCE COMPONENTS (AUXILIARY INPUT)	LB
TAU(3)*			X,Y,Z BODY AXIS TORQUE COMPONENT (AUXILIARY INPUT)	FT-LB
TRM(3)*			X,Y,Z PARENT BODY INERTIAL VELOCITY COMPONENTS TO DETERMINE POSITION RATES DURING TRIM	FT/SEC

*Default = 0

NAME	PORT NO.	DESCRIPTION	UNITS
UCP(3)*		X,Y,Z CREWPERSON BODY AXIS LINEAR VELOCITY VECTOR	FT/SEC
XCP(3)*		X,Y,Z EARTH SYSTEM POSITION VECTOR OF THE CREWPERSON C.G.	FT
WCP(3)*		X,Y,Z CREWPERSON BODY AXIS ANGULAR VELOCITY VECTOR	DEG/SEC
ECP(3)*		EARTH TO CREWPERSON EULER ANGLES (YAW,PITCH,ROLL)	DEG
SCD*		SPINAL COMPRESSION VELOCITY	FT/SEC
SC*		SPINAL COMPRESSION	FT
GX		CREWPERSON X-AXIS LOAD FACTOR	G's
GY		CREWPERSON Y-AXIS LOAD FACTOR	G's
GZ		CREWPERSON Z-AXIS LOAD FACTOR	G's
DR		DYNAMIC RESPONSE	—
FAD(3)		X,Y,Z CREWPERSON BODY AXIS AERODYNAMIC FORCE COMPONENTS	LB
TAD(3)		X,Y,Z CREWPERSON BODY AXIS AERODYNAMIC TORQUE COMPONENTS	FT-LB
WT		WEIGHT OF THE CREWPERSON PLUS CLOTHING AND EQUIPMENT	LB

*These output quantities are states.

NAME	PORT NO.	DESCRIPTION	UNITS
S		AERODYNAMIC REFERENCE AREA	FT^2
B		AERODYNAMIC LATERAL REFERENCE LENGTH	FT
C		AERODYNAMIC LONGITUDINAL REFERENCE LENGTH	FT
CIN(4)		CREWPERSON INERTIA PROPERTIES AFTER SEAT CREWPERSON SEPARATION (IXX,IYY,IZZ,IXZ)	$SLUG\text{-}FT^2$
CX		X-AXIS AERODYNAMIC FORCE COEFFICIENT	—
CY		Y-AXIS AERODYNAMIC FORCE COEFFICIENT	—
CZ		Z-AXIS AERODYNAMIC FORCE COEFFICIENT	—
CL		AERODYNAMIC ROLLING MOMENT COEFFICIENT	—
CM		AERODYNAMIC PITCHING MOMENT COEFFICIENT	—
CN		AERODYNAMIC YAWING MOMENT COEFFICIENT	—
ALP		CREWPERSON ANGLE OF ATTACK	DEG
BET		CREWPERSON SIDESLIP ANGLE	DEG
VM		CREWPERSON MACH NUMBER	—
Q		DYNAMIC PRESSURE	LB/FT^2
ALT		CREWPERSON ALTITUDE	FT
FL		SEAT/CREWPERSON SEPARATION FLAG FOR OUTPUT (1 = SEPARATION)	—

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
COA*			AILERON COMMANDED POSITION	DEG
TCA*			AILERON TIME CONSTANT	SEC
TDA*			AILERON RESPONSE TIME DELAY	SEC
COE*			ELEVATOR COMMANDED POSITION	DEG
TCE*			ELEVATOR TIME CONSTANT	SEC
TDE*			ELEVATOR RESPONSE TIME DELAY	SEC
COR*			RUDDER COMMANDED POSITION	DEG
TCR*			RUDDER TIME DELAY	SEC
TDR*			RUDDER RESPONSE TIME DELAY	SEC
TRM(4)		AE	AIRPLANE CONTROL SURFACE POSITIONS AT TRIM 1) --NOT USED-- 2) AILERON 3) ELEVATOR 4) RUDDER	DEG

*Default values = 0

NAME	PORT NO.	DESCRIPTION	UNITS
AIL*		AILERON DEFLECTION FROM ITS TRIM POSITION	DEG
ELE*		ELEVATOR DEFLECTION FROM ITS TRIM POSITION	DEG
RUD*		RUDDER DEFLECTION FROM ITS TRIM POSITION	DEG

*These output quantities are states

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
TCP			CATAPULT PROPELLANT CONSUMPTION TABLE:	
			PROPELLANT WEB CONSUMED (INDEPENDENT)	IN
			PROPELLANT CONSUMED (DEPENDENT)	SLUGS
SW			FLAG FOR CATAPULT IGNITION (1 = CATAPULT ON)	
UP*			EJECTION DIRECTION FLAG WITH RESPECT TO THE AIRPLANE +1 = UPWARD -1 = DOWNWARD	—
SAP(3)			X,Y,Z SEAT BODY AXIS LINEAR POSITION VECTOR OF THE CATAPULT ATTACHMENT POINT ON THE SEAT	FT
AAP(3)			X,Y,Z AIRPLANE BODY AXIS LINEAR POSITION VECTOR OF THE CATAPULT ATTACHMENT POINT ON THE AIRPLANE	FT
UCL			UNLOADED CATAPULT LENGTH	FT
CSK			CATAPULT STROKE	FT
VI			INITIAL FREE VOLUME	IN3
PA			PISTON AREA	IN^2
PT			TANG RELEASE PRESSURE	LBS/IN^2
CBP			CATAPULT BURST PRESSURE	LBS/IN^2
C			MASS OF TOTAL PROPELLANT	SLUGS
CI			IGNITER PROPELLANT MASS	SLUGS
PMW			PROPELLANT MOLECULAR WEIGHT	LB/LB-MOLE
SK			CATAPULT SPRING CONSTANT	LB/FT

*Default value = 1.

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
CK			CATAPULT DAMPING CONSTANT	LB/FT/SEC
CAM			RATIO OF SPECIFIC HEATS	—
TF			CONSTANT VOLUME FLAME TEMPERATURE	DEG K
C1			FRICTION PROPORTIONALITY CONSTANT	$LB/LB/IN^2$
C2			HEAT LOSS CONSTANT	—

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
B			BURN RATE PROPORTIONALITY CONSTANT	$IN/SEC/(LB/IN^2)$
BXP			BURN RATE EXPONENT	—
TI			CATAPULT TEMPERATURE PRIOR TO IGNITION	DEG K
TDE*			CATAPULT FORCE DECAY TIME	SEC
SRP(3)		SE	X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT	FT
UST(3)		SE	X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT	FT/SEC
EST(3)		SE	EARTH TO SEAT EULER ANGLES (YAW,PITCH,ROLL)	DEG
WST(3)		SE	X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT	DEG/SEC
XAP(3)		AE or SL	X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE AIRPLANE	FT
UAP(3)		AE or SL	X,Y,Z AIRPLANE BODY AXIS LINEAR VELOCITY VECTOR OF THE AIRPLANE CENTER OF GRAVITY	FT/SEC
EAP(3)		AE or SL	EARTH TO AIRPLANE EULER ANGLES (YAW,PITCH,ROLL)	DEG
WAP(3)		AE or SL	X,Y,Z AIRPLANE BODY AXIS ANGULAR VELOCITY VECTOR OF THE AIRPLANE	DEG/SEC

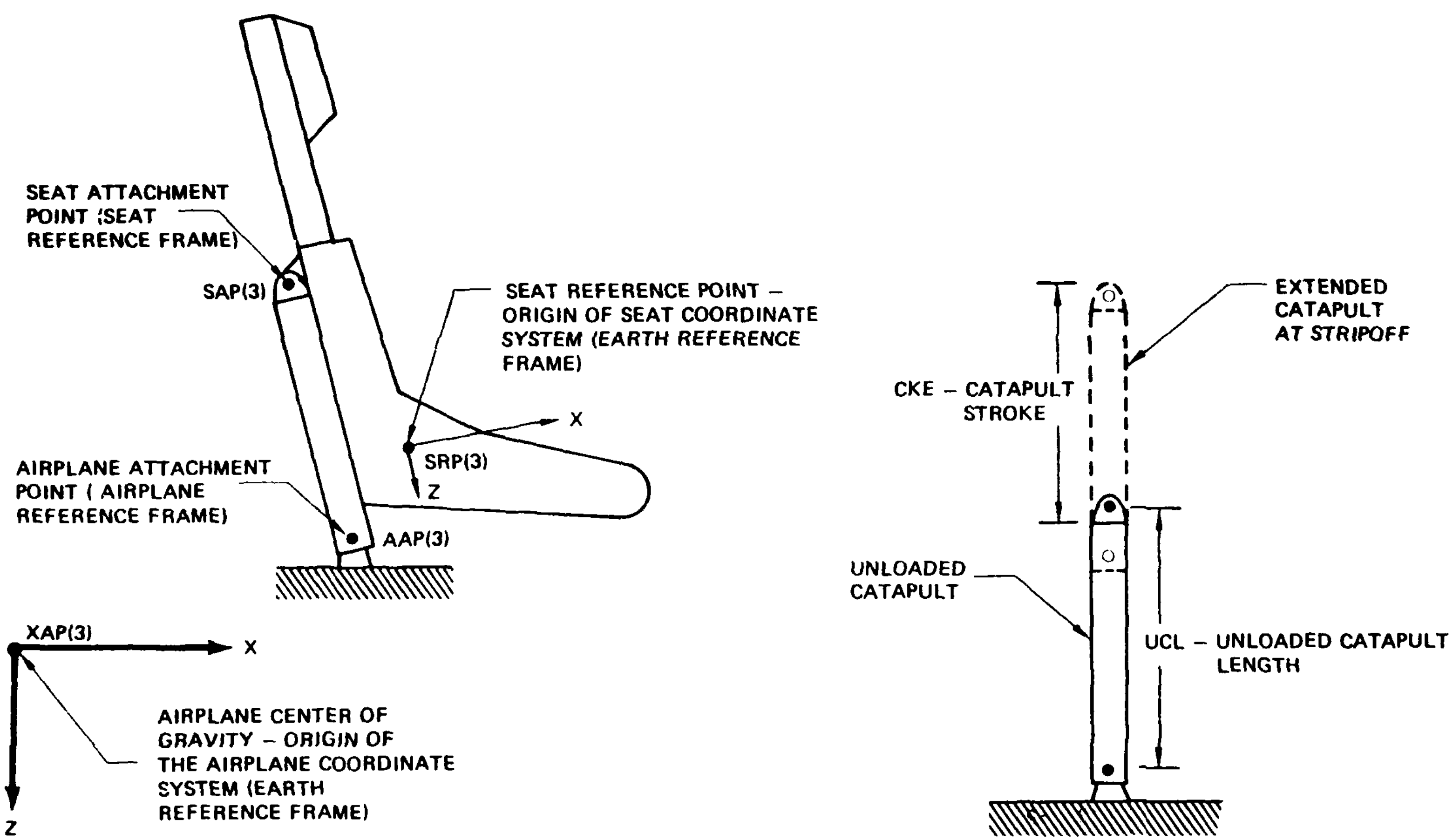
*Default value = 0

NAME	PORT NO.	DESCRIPTION	UNITS
EF*		INTERNAL FRICTION ENERGY	FT-LB
EL*		HEAT LOSS	FT-LB
WK*		CATAPULT WORK	FT-LB
WB*		PROPELLANT WEB CONSUMED	IN
FL		CATAPULT MODE FLAG 0 = PRIOR TO IGNITION 1 = CATAPULT IGNITION 2 = CATAPULT STRIPOFF 3 = CATAPULT OFF	—
FON		STRIPOFF FLAG FOR SUSTAINER ROCKET COMPONENT	
FCA(3)	1	X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS OF THE CATAPULT ON THE AIRPLANE	LB
TCA(3)	1	X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS OF THE CATAPULT ON THE AIRPLANE	FT-LB
F1(3)	1	X,Y,Z SEAT BODY AXIS FORCE COMPONENTS OF THE CATAPULT ON THE SEAT	LB
T1(3)	1	X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS OF THE CATAPULT ON THE SEAT	FT-LB
CF		CATAPULT FORCE MAGNITUDE	LB
CEX		CATAPULT EXTENSION	FT
CV		CATAPULT EXTENSION VELOCITY	FT/SEC
TLØ		INITIAL LENGTH OF THE CATAPULT PRESSURE CHAMBER	IN
PC		CIRCUMFERENCE OF THE CATAPULT PRESSURE CHAMBER	IN

*These output quantities are states.

NAME	PORT NO.	DESCRIPTION	UNITS
R		GAS CONSTANT	FT-LBF/SLUG-K
CVH		CONSTANT VOLUME SPECIFIC HEAT	FT-LBF/SLUG-K
TSO		CATAPULT STRIPOFF TIME	SEC
FSO		CATAPULT FORCE AT STRIPOFF	LB

- STANDARD COMPONENT "CT" CALCULATES THE FORCES AND TORQUES THAT ACT ON THE AIRPLANE AND SEAT
- SUBROUTINE "CAD" COMPUTES *THE CATAPULT FORCE*



CT

Figure 24. Standard Component "CT" Input/Output Overview

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
TBF			DART BRAKING FORCE TABLE:	
			LINE LENGTH (INDEPENDENT)	FT
			BRAKING FORCE (DEPENDENT)	LB
DAP(3)			X,Y,Z AIRPLANE BODY AXIS LINEAR POSITION VECTOR OF THE DART ATTACHMENT POINT	FT
DBA(3)			X,Y,Z SEAT BODY AXIS LINEAR POSITION VECTOR OF THE DEPLOYED DART BRIDLE APEX	FT
XAP(3)		AE or SL	X,Y,Z EARTH LINEAR POSITION VECTOR OF THE AIRPLANE CENTER OF GRAVITY	FT
EAP(3)		AE or SL	EARTH TO AIRPLANE EULER ANGLES (YAW,PITCH,ROLL)	DEG
SRP(3)		SE	X,Y,Z EARTH LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT	FT
EST(3)		SE	EARTH TO SEAT EULER ANGLES (YAW,PITCH,ROLL)	DEG

NAME	PORT NO.	DESCRIPTION	UNITS
F2(3)	1	X,Y,Z SEAT BODY AXIS FORCE COMPONENTS OF THE DART ON THE SEAT	LB
T2(3)	1	X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS OF THE DART ON THE SEAT	FT-LB
FDA(3)	1	X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS OF THE DART ON THE AIRPLANE	LB
TDA(3)	1	X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS OF THE DART ON THE AIRPLANE	FT-LB
DLL		DISTANCE BETWEEN THE BRIDLE APEX AND THE AIRPLANE ATTACHMENT POINT	FT
DBF		DART BRAKING FORCE	LB
SW		DART MODE FLAG 0 = PRIOR TO DART 1 = DART ON 2 = DART OFF	

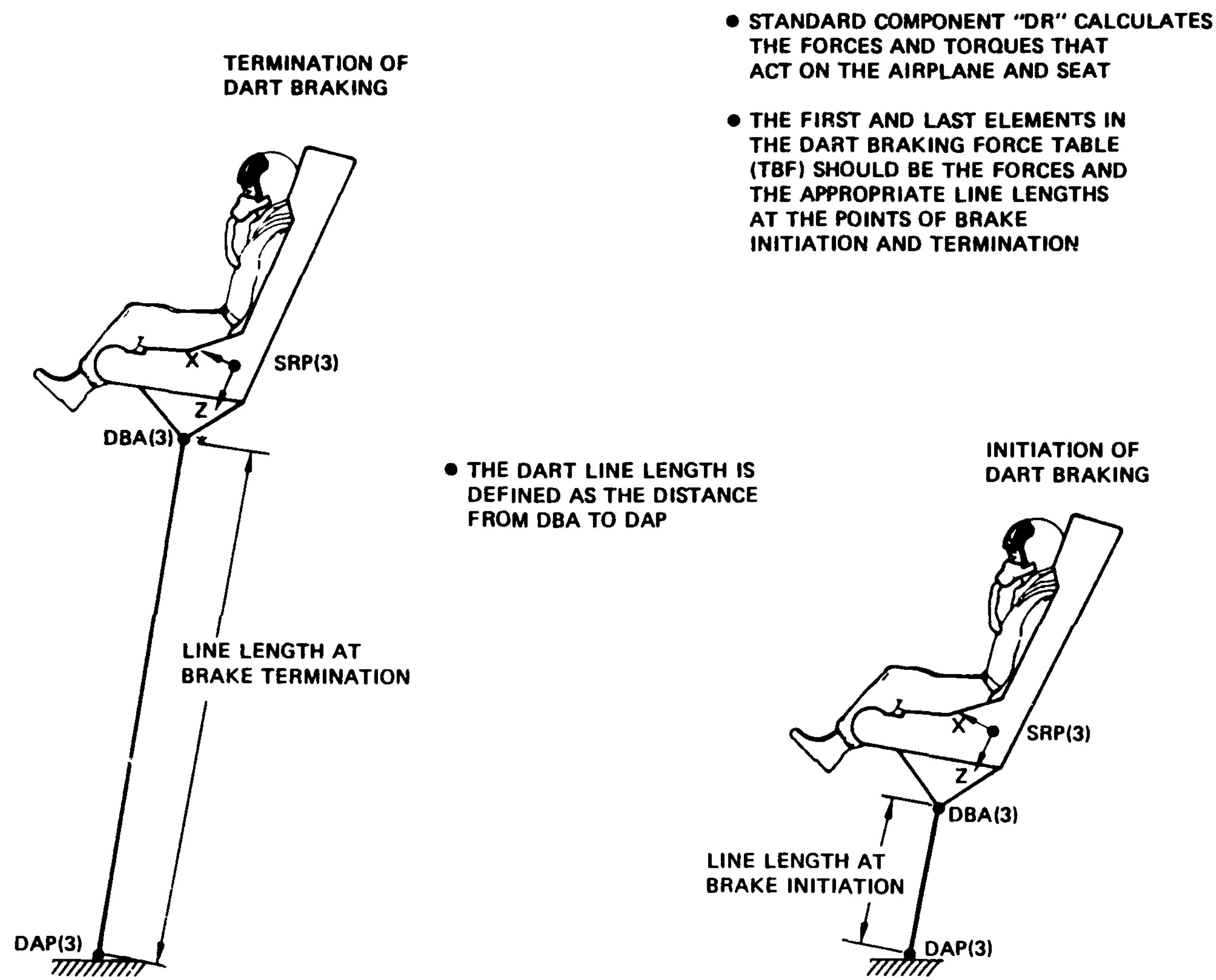


Figure 25. Standard Component "DR" Input/Output Overview

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
TMF			PARACHUTE MORTAR FORCE TABLE: TIME (INDEPENDENT VARIABLE) MORTAR FORCE (DEPENDENT VARIABLE)	 SEC LB
SW			FLAG TO INITIATE MORTAR (1 = ON)	–
UV(3)			X,Y,Z SEAT BODY AXIS MORTAR FORCE DIRECTION UNIT VECTOR ACTING ON THE PARACHUTE PACK	–
XMO(3)			X,Y,Z SEAT BODY AXIS LINEAR POSITION VECTOR OF THE PARACHUTE DEPLOYMENT IMPULSE MOMENT ARM	FT
XYZ(3)			X,Y,Z SEAT BODY AXIS LINEAR POSITION VECTOR OF THE PARACHUTE PACK ATTACHMENT POINT	FT
EA(3)			SEAT TO PARACHUTE PACK ATTACHMENT ATTITUDE EULER ANGLES (YAW, PITCH, ROLL)	DEG
XR			PARACHUTE SHELF LINEAR SPRING CONSTANT	LB/FT
XD			PARACHUTE SHELF LINEAR DAMPING CONSTANT	LB/FT/SEC
ER(3)			X,Y,Z PARACHUTE SHELF ANGULAR SPRING CONSTANTS	FT-LB/DEG
ED(3)			X,Y,Z PARACHUTE SHELF ANGULAR DAMPING CONSTANTS	FT-FT/DEG/SEC
TDE*			TIME DURATION FOR THE MORTAR FORCES AND TORQUES TO DECAY AFTER STRIPOFF	SEC
SRP(3)		SE	X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT	FT

*Default value = 0.

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
UST(3)		SE	X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT	FT/SEC
EST(3)		SE	EARTH TO SEAT EULER ANGLES (YAW,PITCH,ROLL)	DEG
WST(3)		SE	X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT	DEG/SEC
XPP(3)		PC	X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE PACK	FT
UPP(3)		PC	X,Y,Z PARACHUTE PACK BODY AXIS LINEAR VELOCITY VECTOR OF THE PARACHUTE PACK	FT/SEC
EPP(3)		PC	EARTH TO PARACHUTE PACK EULER ANGLES (YAW,PITCH, ROLL)	DEG
WPP(3)		PC	X,Y,Z PARACHUTE PACK BODY AXIS ANGULAR VELOCITY VECTOR OF THE PARACHUTE PACK	DEG/SEC

NAME	PORT NO.	DESCRIPTION	UNITS
FL		PARACHUTE MODE FLAG: 0 = PRIOR TO INITIATION 1 = PARACHUTE INITIATION UP TO LAUNCH 2 = PARACHUTE LAUNCH 3 = FORCES AND TORQUES OFF	–
FMT		PARACHUTE MORTAR FORCE MAGNITUDE	LB
F1	1	X,Y,Z SEAT BODY AXIS FORCE VECTOR ACTING ON THE SEAT	LB
T1	1	X,Y,Z SEAT BODY AXIS TORQUE VECTOR ACTING ON THE SEAT	LB
FPP(3)		X,Y,Z EARTH SYSTEM FORCE VECTOR ACTING ON THE PARACHUTE PACK	LB
TPP(3)		X,Y,Z PARACHUTE PACK BODY AXIS TORQUE VECTOR ACTING ON THE PARACHUTE PACK	FT-LB
TIN		PARACHUTE MORTAR INITIATION TIME	SEC
FSO(3)		X,Y,Z SEAT BODY AXIS FORCE COMPONENTS EXERTED ON THE SEAT AT STRIPOFF	LB
TSO(3)		X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS EXERTED ON THE SEAT AT STRIPOFF	FT-LB
FPO(3)		X,Y,Z EARTH SYSTEM FORCE COMPONENTS EXERTED ON THE SEAT AT STRIPOFF (LB)	LB
TPO(3)		X,Y,Z PARACHUTE PACK BODY AXIS TORQUE COMPONENTS EXERTED ON THE SEAT AT STRIPOFF	FT-LB
TRM(3)		X,Y,Z SEAT INERTIAL VELOCITY COMPONENTS TO PASS TO THE PARACHUTE COMPONENT DURING TRIM	FT/SEC

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
TCW			STRETCHED CANOPY WEIGHT TABLE:	
			STRETCHED LENGTH (INDEPENDENT)	FT
			STRETCHED WEIGHT (DEPENDENT)	LB
OFF*			FLAG TO SEVER LINES 0 = LINES ATTACHED 1 = LINES SEVERED	
BLI			NUMBER OF BRIDLE LINES	—
APX(3)*			X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE BRIDLE APEX	FT
AP1(3)			X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE FIRST BRIDLE LINE ATTACHMENT POINT	FT
AP2(3)*			X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE SECOND BRIDLE LINE ATTACHMENT POINT	FT
AP3(3)*			X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE THIRD BRIDLE LINE ATTACHMENT POINT	FT
AP4(3)*			X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE FOURTH BRIDLE LINE ATTACHMENT POINT	FT
FTR			PARACHUTE LINE MULTIPLICATION FACTOR	—
FSO			CANOPY STRIPOUT FORCE	LB
ULL			PARACHUTE SUSPENSION LINE ULTIMATE LOAD	IN/IN
ULS			PARACHUTE SUSPENSION LINE ULTIMATE STRAIN	IN/IN

*Default value = 0

LI

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
GOR			NUMBER OF PARACHUTE GORES	–
TYP			TYPE OF PARACHUTE (1 = DRAG 2 = RECOVERY)	–
FL		MP or GP	MORTAR MODE FLAG 0 = PRIOR TO INITIATION 1 = INITIATION 2 = LAUNCH	–
XDO(3)		SE or CE	X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE DECELERATED OBJECT	FT
UDO(3)		SE or CE	X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR VELOCITY VECTOR	FT/SEC
EDO(3)		SE or CE	EARTH TO DECELERATED OBJECT EULER ANGLES (YAW, PITCH, ROLL)	DEG
WDO(3)		SE or CE	X,Y,Z DECELERATED OBJECT BODY AXIS ANGULAR VELOCITY COMPONENTS	DEG/SEC
XPP(3)		PC	X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE PACK	FT
UPP(3)		PC	X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE PARACHUTE PACK	FT/SEC
EPP(3)		PC	EARTH TO PARACHUTE PACK EULER ANGLES (YAW, PITCH, ROLL)	DEG
XPC(3)		PC	X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE CANOPY	FT
UPC(3)		PC	X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE PARACHUTE CANOPY	FT/SEC

NAME	PORT NO.	DESCRIPTION	UNITS
EC*		CREEP STRAIN IN PARACHUTE LINES	IN/IN
TF*		TIME DURATION OF A NON-ZERO LOAD ON THE LINES	SEC
FLA		PARACHUTE PHASE 0 = PRIOR TO INITIATION 1 = INITIATION 2 = LAUNCH 3 = LINESTRETCH 4 = LINES SEVERED	–
SW1		FLAG SET WHEN PARACHUTE IS BEHIND THE BRIDLE APEX (1 = BEHIND)	–
FDO(3)		X,Y,Z DECELERATED OBJECT BODY AXIS FORCE COMPONENTS	LB
TDO(3)		X,Y,Z DECELERATED OBJECT BODY AXIS TORQUE COMPONENTS	FT-LB
FLP(3)		X,Y,Z EARTH SYSTEM FORCE COMPONENTS ACTING ON THE PARACHUTE CANOPY	LB
FAP(3)		X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR OF THE FORCE APPLICATION POINT	FT
VAP(3)		X,Y,Z EARTH SYSTEM VELOCITY COMPONENTS OF THE FORCE APPLICATION POINT	FT/SEC
FLL		LINE LOAD	LB
ELM		MAXIMUM STRAIN EXPERIENCED BY THE PARACHUTE LINE DURING ITS LOADING HISTORY	IN/IN

*These output quantities are states.

NAME	PORT NO.	DESCRIPTION	UNITS
ELC		MAXIMUM STRAIN EXPERIENCED BY THE PARACHUTE LINE DURING THE CURRENT LOADING CYCLE ONLY	IN/IN
DEM		MAXIMUM NEGATIVE STRAIN RATE EXPERIENCED BY THE PARACHUTE LINE DURING ITS LOADING HISTORY	1/SEC
RMN		MAXIMUM NEGATIVE STRAIN RATE EXPERIENCED BY THE PARACHUTE LINE DURING THE CURRENT UN-LOADING CYCLE ONLY	1/SEC
DIS		THE DISTANCE FROM THE ORIGIN OF THE DECELERATED OBJECT TO THE BRIDLE APEX	FT
CON(4)		COEFFICIENTS IN THE EQUATION FOR THE PLANE FORMED BY THE BRIDLE ATTACHMENT POINTS	—
TCG(20)		STRETCHED CANOPY CENTER OF GRAVITY LOCATION ARRAY	FT
UVL(3)		PARACHUTE LINE UNIT VECTOR	—
RL		PARACHUTE LINE LENGTH	FT
RLO		UNLOADED PARACHUTE LINE LENGTH	FT
VL		RATE OF CHANGE OF LINE LENGTH	FT/SEC
VCG		VELOCITY OF THE STRETCHED CANOPY CENTER OF GRAVITY ALONG THE LINES	FT/SEC
PCG		STRETCHED CANOPY CENTER OF GRAVITY MEASURED ALONG THE PARACHUTE LINE FROM THE PARACHUTE PACK	FT
CWT		WEIGHT OF THE CANOPY PULLED FROM THE PARACHUTE PACK	LB

NAME	PORT NO.	DESCRIPTION	UNITS
TPE		TYPE OF PARACHUTE (1 = DRAG 2 = RECOVERY)	—
PVL		PREVIOUS TIMESTEP LINE VELOCITY	FT/SEC
TLS		TIME AT LINESTRETCH	SEC
VLS		RATE OF CHANGE OF LINE LENGTH AT LINESTRETCH	FT/SEC

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	MP UNITS
C1			FRICTION PROPORTIONALITY CONSTANT	$LB/LB/IM^2$
C2			HEAT LOSS CONSTANT	-
B			BURN RATE PROPORTIONALITY CONSTANT	$IN/SEC/(LB/IN^2)$
BXP			BURN RATE EXPONENT	-
TI			MORTAR TEMPERATURE PRIOR TO IGNITION	DEG K
TDE*			MORTAR FORCE DECAY TIME	SEC
SRP(3)		SE	X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT	FT
UST(3)		SE	X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT	FT/SEC
EST(3)		SE	EARTH TO SEAT EULER ANGLES	DEG
WST(3)		SE	X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT	DEG/SEC
XPP(3)		PC	X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE PACK	FT
UPP(3)		PC	X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE PARACHUTE PACK	FT/SEC
EPP(3)			EARTH TO PARACHUTE PACK EULER ANGLES	DEG
WPP(3)			X,Y,Z PARACHUTE PACK BODY AXIS ANGULAR VELOCITY VECTOR OF THE PARACHUTE PACK	DEG/SEC

*Default value = 0

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	MP UNITS
SW			FLAG TO INITIATE THE MORTAR (1 = ON)	-
XYZ(3)			X,Y,Z SEAT BODY AXIS LINEAR POSITION VECTOR OF THE PARACHUTE PACK ATTACHMENT POINT ON THE SEAT	FT
EA(3)			SEAT TO PARACHUTE PACK ATTACHMENT EULER ANGLES	DEG
XR			PARACHUTE SHELF LINEAR SPRING CONSTANT	LB/FT
XD			PARACHUTE SHELF LINEAR DAMPING CONSTANT	LB/FT/SEC
ER(3)			X,Y,Z PARACHUTE SHELF ANGULAR SPRING CONSTANT	FT-LB/DEG
ED(3)			X,Y,Z PARACHUTE SHELF ANGULAR DAMPING CONSTANT	FT-LB/DEG/SEC
UV(3)			X,Y,Z SEAT BODY AXIS MORTAR FORCE UNIT VECTOR	-
CSK			MORTAR STROKE	FT
VI			INITIAL FREE VOLUME	IN^3
PA			PISTON AREA	IN^2
PT			TANG RELEASE PRESSURE	LB/IN^2
CBP			MORTAR BURST PRESSURE	LB/IN^2
C			MASS OF TOTAL PROPELLANT	SLUGS
CI			IGNITER PROPELLANT MASS	SLUGS
PMW			PROPELLANT MOLECULAR WEIGHT	LB/LB-MOLE
GAM			RATIO OF SPECIFIC HEATS	-
TF			CONSTANT VOLUME FLAME TEMPERATURE	DEG K

NAME	PORT NO.	DESCRIPTION	UNITS
EF*		INTERNAL FRICTION ENERGY	FT-LB
EL*		HEAT LOSS ENERGY	FT-LB
WK*		MORTAR WORK	FT-LB
WB*		PROPELLANT WEB BURNED	IN
FL		MORTAR MODE FLAG 0 = PRIOR TO INITIATION 1 = INITIATION 2 = LAUNCH 3 = MORTAR OFF	-
F1(3)	1	X,Y,Z SEAT BODY AXIS FORCE COMPONENTS OF THE MORTAR AND RESTRAINTS ON THE SEAT	LB
TI(3)	1	X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS OF THE MORTAR AND RESTRAINTS ON THE SEAT	FT-LB
FPP(3)		X,Y,Z EARTH SYSTEM FORCE COMPONENTS OF THE MORTAR AND RESTRAINTS ON THE PARACHUTE PACK	LB
TPP(3)		X,Y,Z PARACHUTE PACK BODY AXIS TORQUE COMPONENTS OF THE MORTAR AND RESTRAINTS ON THE PARACHUTE PACK	FT-LB
FM		MORTAR FORCE MAGNITUDE	LB
EXM		MORTAR EXTENSION	FT
VM		MORTAR EXTENSION VELOCITY	FT/SEC
TSO		MORTAR STRIPOFF TIME	SEC
FSO		FORCE AT MORTAR STRIPOFF	LB
TRM(3)		X,Y,Z SEAT EARTH SYSTEM VELOCITY COMPONENTS TO PASS TO THE PARACHUTE COMPONENT DURING TRIM	FT/SEC

*These output quantities are states

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
STI			INFLATED PARACHUTE DRAG AREA	FT^2
RCS			CIRCUMFERENCE OF THE FILLED CANOPY PLUS ONE QUARTER OF THAT DISTANCE	FT
RFM			REEF MODE FLAG 0 = CHUTE NOT REEFED 1 = TIME OF DISREEF SET AT PARACHUTE INITIATION 2 = TIME OF DISREEF SET AT LINESTRETCH	-
RFD			REEF DELAY TIME	SEC
RFS			PRODUCT OF REFERENCE AREA AND TANGENT FORCE COEFFICIENT WHEN REEFED	FT^2
B			CONSTANT USED IN THE EQUATION THAT CALCULATES SCD OF THE REEFED PARACHUTE	-
CI			CONSTANT USED IN THE EQUATION TO COMPUTE THE CANOPY INFLATION TIME	-
CT(3)			CONSTANTS USED IN THE EQUATION THAT CALCULATES THE TANGENTIAL DRAG AREA	-
CN(3)			CONSTANTS USED IN THE EQUATION THAT CALCULATES THE NORMAL DRAG AREA	-
CM(2)			CONSTANTS USED IN THE MACH EFFECTS EQUATION	-
FD			WAKE TO FREE STREAM RATIO	
PWT			TOTAL WEIGHT OF THE PARACHUTE PACK	LB
PMI(3)			PARACHUTE PACK MOMENTS OF INERTIA (IXX,IYY,IZZ)	SLUG-FT^2

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
PPI(3)			PARACHUTE PACK PRODUCTS OF INERTIA (IXY,IXZ,IYZ)	SLUG-FT^2
TEM*			TIME DURATION FOR PARACHUTE EMERGENCE	SEC
CSP**			PARACHUTE CANOPY SPRING CONSTANT	LB/FT
CDP***			PARACHUTE CANOPY DAMPING CONSTANT	LB/FT/SEC
FLA		LI	PARACHUTE MODE FLAG 0 = PRIOR TO INITIATION 1 = INITIATION 2 = LAUNCH 3 = LINESTRETCH 4 = LINES SEVERED	-
FLP(3)		LI	X,Y,Z EARTH SYSTEM FORCE COMPONENTS ACTING ON THE PARACHUTE FROM THE LINES	LB
FPP(3)		GP or MP	X,Y,Z EARTH SYSTEM FORCE COMPONENTS ACTING ON THE PACK FROM THE RESTRAINTS AND MORTAR	LB
TPP(3)		GP or MP	X,Y,Z PARACHUTE PACK BODY AXIS TORQUE COMPONENTS ACTING ON THE PACK FROM THE RESTRAINTS	FT-LB
VAP		LI	X,Y,Z EARTH SYSTEM VELOCITY COMPONENTS OF THE FORCE APPLICATION POINT	FT/SEC
UVL(3)		LI	EARTH SYSTEM PARACHUTE LINE UNIT VECTOR	-

*Default value = 0
**Default value = 2000.
***Default value = 14.

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
RL		LI	PARACHUTE LINE LENGTH	FT
VCG(3)		LI	VELOCITY OF THE CANOPY CENTER OF GRAVITY ALONG THE PARACHUTE LINES	FT/SEC
PCG		LI	STRETCHED CANOPY CENTER OF GRAVITY MEASURED ALONG THE PARACHUTE LINE FROM THE PARACHUTE PACK	FT
CWT		LI	WEIGHT OF THE CANOPY DRAWN FROM THE PACK	LB
TPE		LI	TYPE OF PARACHUTE 1 = DRAG 2 = RECOVERY	-
TRM(3)		GP or MP	X,Y,Z PARENT BODY EARTH SYSTEM VELOCITY COMPONENTS TO DETERMINE THE POSITION RATES DURING TRIM	FT/SEC

NAME	PORT NO.	DESCRIPTION	UNITS
UPP(3)*		X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE PARACHUTE PACK CENTER OF GRAVITY	FT/SEC
XPP(3)*		X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE PACK CENTER OF GRAVITY	FT
WPP(3)*		X,Y,Z PARACHUTE PACK BODY AXIS ANGULAR VELOCITY VECTOR	DEG/SEC
EPP(3)*		EARTH TO PARACHUTE PACK EULER ANGLES (YAW, PITCH, ROLL)	DEG
UPC(3)*		X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE PARACHUTE CANOPY	FT/SEC
XPC(3)*		X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE CANOPY	FT
PHA		PARACHUTE PHASE 1 = PRIOR TO PARACHUTE LAUNCH 2 = FROM LAUNCH UP TO LINE-STRETCH 3 = AFTER LINESTRETCH	-

*These output quantities are states

NAME	PORT NO.	DESCRIPTION	UNITS
SW		FLAG TO INDICATE PARACHUTE AERODYNAMIC CALCULATION MODE: 0 = PRIOR TO LAUNCH 1 = FROM PARACHUTE LAUNCH TO LINESTRETCH 2 = DURING INFLATION 3 = DURING REEFING 4 = AFTER REEFING 5 = PARACHUTE INFLATED	-
FLI(3)*		X,Y,Z EARTH SYSTEM AERODYNAMIC LIFT COMPONENTS	LB
FDR(3)*		X,Y,Z EARTH SYSTEM AERODYNAMIC DRAG COMPONENTS	LB
FMA(3)		X,Y,Z EARTH SYSTEM FORCE COMPONENTS ACTING ON THE CANOPY DUE TO AIR MASS ACQUISITION FORCE	LB
RM		RADIUS OF THE SPHERE REPRESENTING THE INFLATED CANOPY	FT
VOL		VOLUME OF THE FILLED CANOPY	FT^3
TLA		PARACHUTE LAUNCH TIME OR LINE SEVERING TIME	SEC
TLS		LINESTRETCH TIME	SEC
TDS		TIME AT WHICH DISREEF OCCURS	SEC

*Acting on the pack before linestretch
Acting on the canopy after linestretch

NAME	PORT NO.	DESCRIPTION	UNITS
DTI		PARACHUTE CANOPY INFLA-TION TIME	SEC
TDU		TIME DURATION OF REEFED PARACHUTE	SEC
TRF		TIME AT WHICH THE CHUTE IS REEFED	SEC

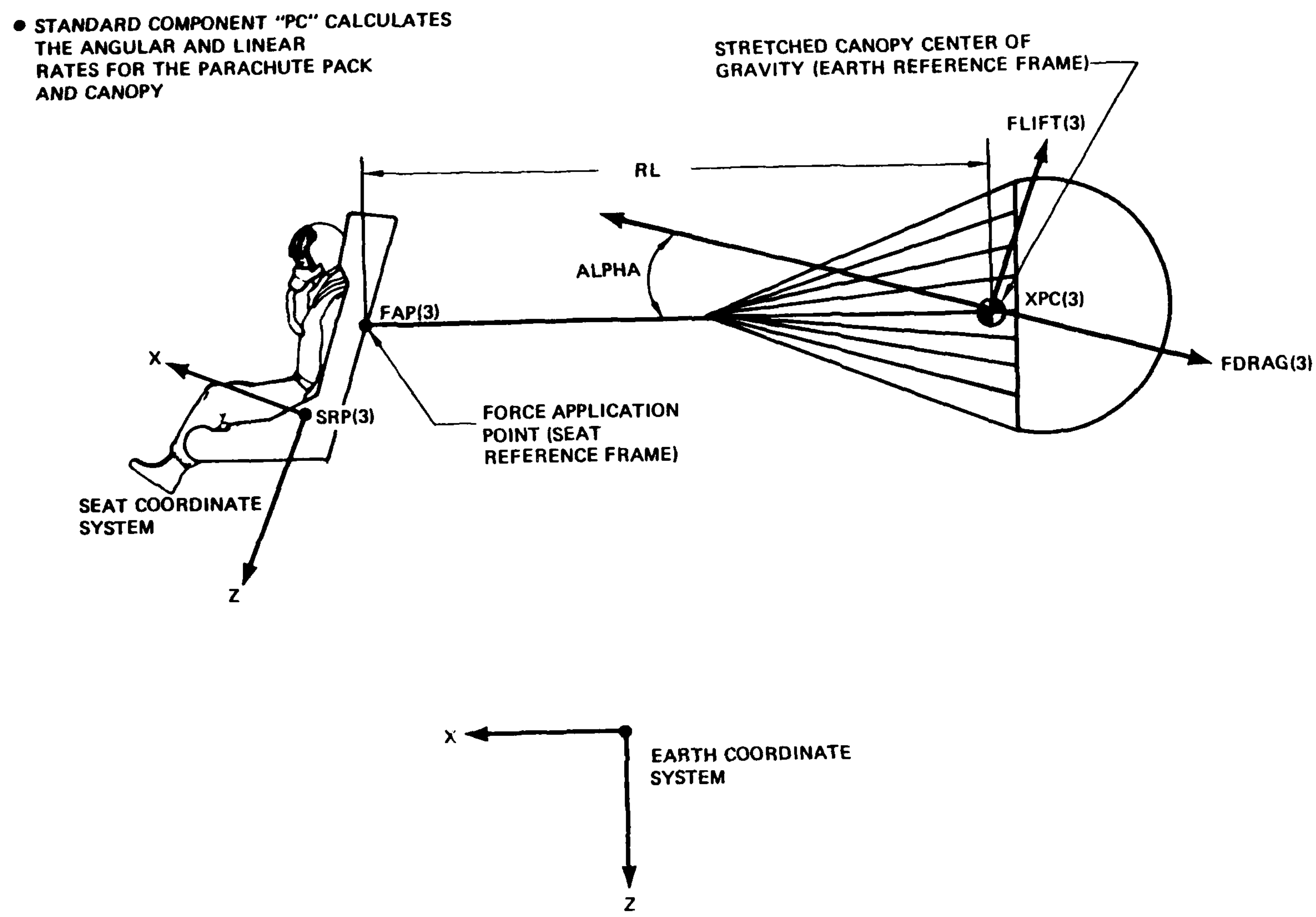


Figure 26. Standard Component "PC" Input/Output Overview

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
BL1(3)			X,Y,Z SEAT AXIS POSITION VECTOR OF THE RIGHT LOWER BLOCK	FT
BL2(3)			X,Y,Z SEAT AXIS POSITION VECTOR OF THE RIGHT MIDDLE BLOCK	FT
BL3(3)			X,Y,Z SEAT AXIS POSITION VECTOR OF THE RIGHT UPPER BLOCK	FT
BL4(3)			X,Y,Z SEAT AXIS POSITION VECTOR OF THE LEFT LOWER BLOCK	FT
BL5(3)			X,Y,Z SEAT AXIS POSITION VECTOR OF THE LEFT MIDDLE BLOCK	FT
BL6(3)			X,Y,Z SEAT AXIS POSITION VECTOR OF THE LEFT UPPER BLOCK	FT
UP			EJECTION DIRECTION FLAG +1 = UPWARD WRT THE AIRPLANE -1 = DOWNWARD WRT THE AIRPLANE	–
RLR			RIGHT RAIL Z COORDINATE OF THE END OF THE RIGHT RAIL	FT
XRR(3)			X,Y,Z AIRPLANE POSITION VECTOR OF THE ORIGIN OF THE RIGHT RAIL COORDINATE SYSTEM	FT
RLL			LEFT RAIL Z COORDINATE OF THE END OF THE LEFT RAIL	FT

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
XRL(3)			X,Y,Z AIRPLANE POSITION VECTOR OF THE ORIGIN OF THE LEFT RAIL COORDINATE SYSTEM	FT
ERL(3)			AIRPLANE TO RAILS EULER ANGLES (YAW,PITCH,ROLL)	DEG
SPR(2)			X,Y RAIL SPRING CONSTANTS	LB/FT
DPG (2)			X,Y RAIL DAMPING CONSTANTS	LB/FT/SEC
SBF			SLIDER BLOCK FRICTION COEFFICIENT	-
ZTS			RIGHT RAIL AXIS Z COORDINATE OF THE KEY BLOCK AT TRIP SWITCH CONTACT	FT
BTS			TRIP SWITCH KEY BLOCK NUMBER 1 = BOTTOM RIGHT BLOCK 2 = MIDDLE RIGHT BLOCK 3 = TOP RIGHT BLOCK	
CPT(3)			X,Y,Z AIRPLANE POSITION VECTOR OF THE CRITICAL CLEARANCE POINT	FT
SRP(3)		SE	X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT	FT
UST(3)		SE	X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT	FT/SEC
EST(3)		SE	EARTH TO SEAT EULER ANGLES (YAW,PITCH,ROLL)	DEG
WST(3)		SE	X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT	DEG/SEC
XAP(3)		AE or SL	X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE AIRPLANE	FT

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
UAP(3)		AE or SL	X,Y,Z AIRPLANE BODY AXIS LINEAR VELOCITY VECTOR OF THE AIRPLANE	FT/SEC
EAP(3)		AE or SL	EARTH TO AIRPLANE EULER ANGLES (YAW, PITCH, ROLL)	DEG
WAP(3)		AE or SL	X,Y,Z AIRPLANE BODY AXIS ANGULAR VELOCITY VECTOR	DEG/SEC

RL

NAME	PORT NO.	DESCRIPTION	UNITS
F2(3)	1	X,Y,Z SEAT BODY AXIS FORCE COMPONENTS ON THE SEAT FROM THE RAILS	LB
T2(3)	1	X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS ON THE SEAT FROM THE RAILS	FT-LB
FR1(3)	1	X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS ON THE AIRPLANE FROM THE RAILS	LB
TR1(3)	1	X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS ON THE AIRPLANE FROM THE RAILS	FT-LB
FL		STROKE FLAG - 0 = GUIDED 1 = UNGUIDED	
FTS		TRIP SWITCH CONTACT FLAG (1 = ON)	-
TTS		TRIP SWITCH CONTACT TIME	SEC
OFF		SEAT/RAIL SEPARATION FLAG (1 = SEPARATION)	-
DSA (3,3)		SEAT TO AIRPLANE DIRECTION COSINE MATRIX	-
SRA(3)		X,Y,Z AIRPLANE COORDINATE SYSTEM LINEAR POSITION VECTOR OF THE SRP	FT
DIS		DISTANCE FROM THE CRITICAL POINT TO THE SEAT REFERENCE POINT	FT
TM(3)		X,Y,Z EARTH VELOCITY COMPONENTS OF THE VEHICLE TO PASS TO THE SEAT DURING TRIM	FT/SEC

- STANDARD COMPONENT "RL" CALCULATES THE FORCES AND TORQUES THAT ACT ON THE AIRPLANE AND SEAT FROM THE RAILS
- THE FORCES AND TORQUES THAT ARE CALCULATED AIRE ARE A FUNCTION OF THE LINEAR DAIS DISPLACEMENT OF THE BLOCKS FROM THE RAILS
- EACH RAIL HAS A COORDINATE SYSTEM ATTACHED TO IT

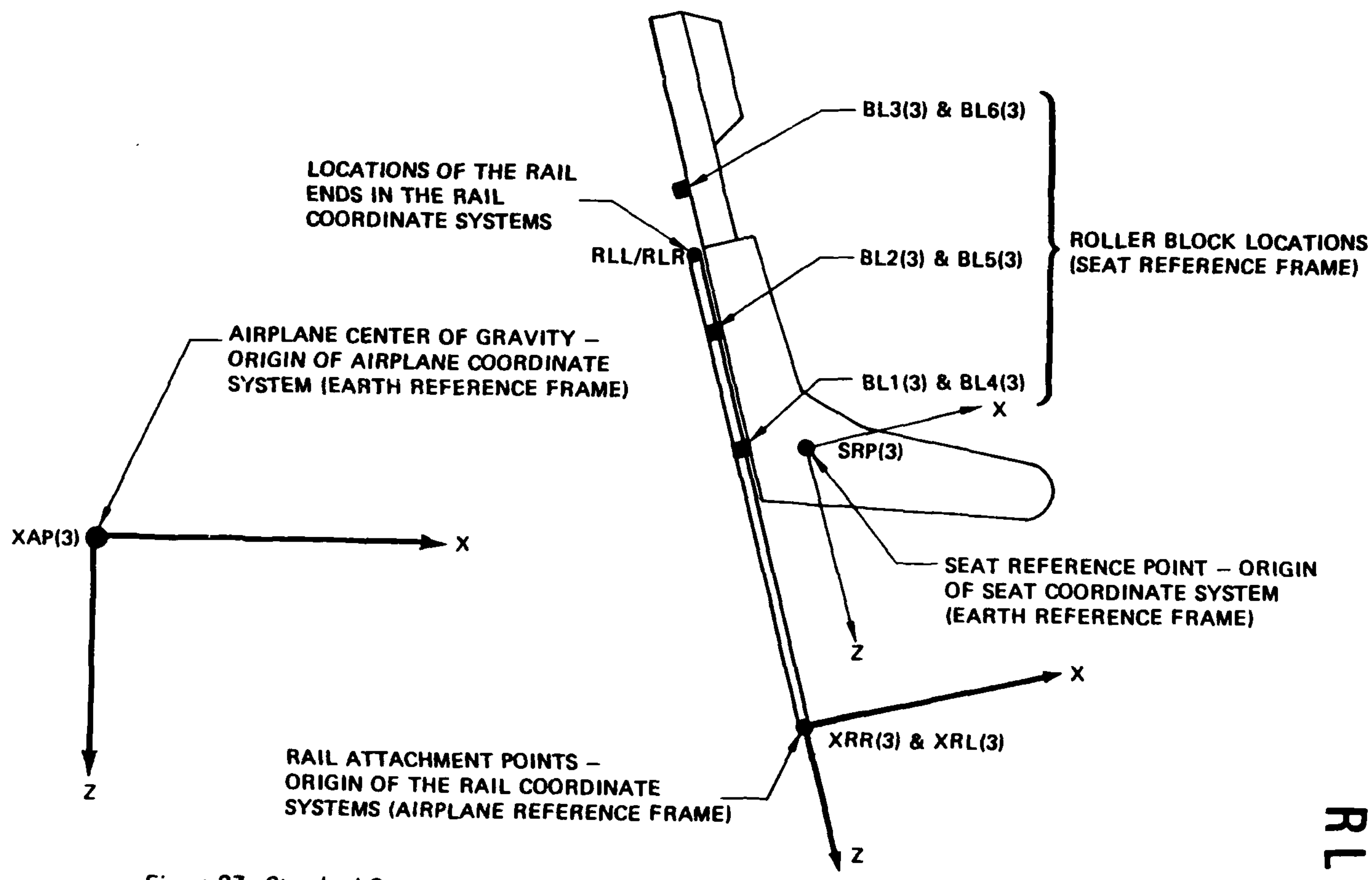


Figure 27. Standard Component "RL" Input/Output Overview

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
FL			FLAG TO RELEASE ATTACHED BODY (1 = RELEASE)	-
XYZ(3)			X,Y,Z PARENT BODY AXIS LINEAR POSITION VECTOR OF THE ATTACHMENT POINT	FT
EA(3)			PARENT BODY TO ATTACHED BODY ATTACHMENT POSITION EULER ANGLES (YAW,PITCH, ROLL)	DEG
XPB(3)			X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARENT BODY	FT
UPB(3)			X,Y,Z PARENT BODY AXIS LINEAR VELOCITY VECTOR OF THE PARENT BODY	FT/SEC
EPB(3)			EARTH TO PARENT BODY EULER ANGLES (YAW,PITCH, ROLL)	DEG
WPB(3)			X,Y,Z BODY AXIS ANGULAR VELOCITY VECTOR OF THE PARENT BODY	DEG/SEC
XAB(3)			X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE ATTACHED BODY	FT
UAB(3)			X,Y,Z ATTACHED BODY AXIS LINEAR VELOCITY VECTOR OF THE ATTACHED BODY	FT/SEC
EAB(3)			EARTH TO ATTACHED BODY EULER ANGLES (YAW,PITCH,ROLL)	DEG
WAB(3)			X,Y,Z ATTACHED BODY AXIS ANGULAR VELOCITY VECTOR OF THE ATTACHED BODY	DEG/SEC

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
XR			LINEAR SPRING CONSTANT	LB/FT
XD			LINEAR DAMPING CONSTANT	LB/FT/SEC
ER(3)			X,Y,Z ANGULAR SPRING CONSTANT	FT-LB/DEG
ED(3)			X,Y,Z ANGULAR DAMPING CONSTANT	FT-LB/DEG/SEC

NAME	PORT NO.	DESCRIPTION	UNITS
FPB(3)		X,Y,Z PARENT BODY AXIS FORCE VECTOR	LB
TPB(3)		X,Y,Z PARENT BODY AXIS TORQUE VECTOR	FT-LB
FAB(3)		X,Y,Z ATTACHED BODY AXIS FORCE VECTOR	LB'
TAB(3)		X,Y,Z ATTACHED BODY AXIS TORQUE VECTOR	FT-LB
TRM(3)		X,Y,Z PARENT BODY EARTH SYSTEM VELOCITY COMPONENTS TO PASS TO THE ATTACHED BODY	FT/SEC

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
F1(3)*	1 TO 9		X,Y,Z SEAT BODY AXIS FORCE COMPONENTS GENERATED BY A PYROTECHNIC DEVICE	LBS
T1(3)*	1 TO 9		X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS ABOUT THE SRP GENERATED BY A PYROTECHNIC DEVICE	FT-LBS
F2(3)*	1 TO 9		X,Y,Z SEAT BODY AXIS FORCE COMPONENTS GENERATED BY A NON-PYROTECHNIC DEVICE	LBS
T2(3)*	1 TO 9		X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS ABOUT THE SRP GENERATED BY A NON-PYROTECHNIC DEVICE	FT-LBS
CW		WB	COMPOSITE WEIGHT OF THE SEAT	LBS
CCG(3)		WB	X,Y,Z SEAT AXIS SYSTEM COMPOSITE CENTER OF GRAVITY	FT
CMI(3)		WB	MOMENT OF INERTIA VECTOR ABOUT THE SEAT REFERENCE POINT FOR THE COMPOSITE SEAT (IXY,IYY,IZZ)	SLUG-FT2
CPI(3)		WB	PRODUCT OF INERTIA VECTOR ABOUT THE SEAT REFERENCE POINT FOR THE COMPOSITE SEAT (IXY,IXZ,IYZ)	SLUG-FT2
TM(3)		RL	X,Y,Z VEHICLE EARTH VELOCITY COMPONENTS TO DETERMINE THE POSITION RATE DURING TRIM	FT/SEC

* Default = 0.

NAME	PORT NO.	DESCRIPTION	UNITS
UST(3)*		X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE SEAT REFERENCE POINT	FT/SEC
SRP(3)*		X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT REFERENCE POINT	FT
WST(3)*		X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT	DEG/SEC
EST(3)*		EARTH TO SEAT EULER ANGLES (YAW,PITCH,ROLL)	DEG
ALT		SEAT ALTITUDE	FT

* These output quantities are states.

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
UD(3)*			X,Y,Z SLED SYSTEM LINEAR VELOCITY RATE VECTOR	FT/SEC/SEC
WD(3)*			X,Y,Z SLED SYSTEM ANGULAR VELOCITY RATE VECTOR	DEG/SEC/SEC

*Default value = 0.

NAME	PORT NO.	DESCRIPTION	UNITS
UAP(3)*		X,Y,Z SLED BODY AXIS LINEAR VELOCITY COMPONENTS	FT/SEC
XAP(3)*		X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SLED	FT
WAP(3)*		X,Y,Z SLED BODY AXIS ANGULAR VELOCITY COMPONENTS	DEG/SEC
EAP(3)*		EARTH TO SLED EULER ANGLES (YAW,PITCH,ROLL)	DEG

*These output quantities are states.

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
TRF			ROCKET TABLE: TIME (INDEPENDENT) FORCE (DEPENDENT)	 SEC LBS
TMA			MECHANICAL ADVANTAGE TABLE: GIMBAL ANGLE (INDEPENDENT) MECHANICAL ADVANTAGE (DEPENDENT)	 DEG -
TST			SPRING MOMENT TABLE: GIMBAL ANGLE (INDEPENDENT) SPRING TORQUE (DEPENDENT)	 DEG FT-LBS
FL			STAPAC IGNITION FLAG (1 = STAPAC ON)	
YPR			STAPAC APPLICATION FLAG 1 = YAW STAPAC 2 = PITCH STAPAC 3 = ROLL STAPAC	
AVW			ANGULAR VELOCITY OF GYROSCOPE WHEEL	DEG/SEC
WMI			MOMENT OF INERTIA OF THE WHEEL ABOUT ITS SPIN AXIS	SLUG-FT^2
SMI			MOMENT OF INERTIA OF THE SYSTEM LESS THE ROCKET ABOUT THE GIMBAL AXIS	SLUG-FT^2
RII			MOMENT OF INERTIA OF THE ROCKET PRIOR TO IGNITION	SLUG-FT^2
RIF			MOMENT OF INERTIA OF THE ROCKET AFTER BURNOUT	SLUG-FT^2
XR(3)			X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE ROCKET NOZZLE	FT

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
UV(3)**			X,Y,Z ROCKET FORCE UNIT VECTOR IN THE ROCKET COORDINATE SYSTEM	-
GSA			GIMBAL MOTION STOP IN THE NEGATIVE ROLL DIRECTION (MEASURED FROM THE CAGED POSITION)	DEG
GSF			GIMBAL MOTION STOP IN THE POSITIVE ROLL DIRECTION (MEASURED FROM THE CAGED POSITION)	DEG
SPR			GIMBAL STOP ANGULAR RIGIDITY	FT-LB/DEG
DPG			GIMBAL STOP ANGULAR DAMPING	FT-LB/DEG/SEC
FMT			LOAD AT MAXIMUM FRICTION	LBS
TMX			MAXIMUM FRICTION	FT-LB
TNF			FRICTION AT NO THRUST	FT-LB
TOS			THRUSTLINE OFFSET	FT
TSU*			GYROSCOPE WHEEL SPINUP TIME (SEC)	SEC
GMA*			GIMBAL ANGULAR VELOCITY AT MAXIMUM FRICTION	DEG/SEC
WST(3)		SE	X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF THE SEAT	DEG/SEC

*Defaults = 0.
**Defaults: UV(1) = 0.
UV(2) = 0.
UV(3) = -1.

NAME	PORT NO.	DESCRIPTION	UNITS
WG*		GIMBAL SYSTEM X-AXIS ANGULAR VELOCITY	DEG/SEC
ESG(3)*		SEAT TO GIMBAL EULER ANGLES (YAW,PITCH,ROLL)	DEG
ESR(3)*		SEAT TO ROCKET EULER ANGLES (YAW,PITCH,ROLL)	DEG
PHA		STAPAC OPERATIONAL PHASE 0 = BEFORE IGNITION 1 = STAPAC IGNITION 2 = STAPAC BURNOUT	
F1(3)	1	X,Y,Z SEAT BODY AXIS FORCE COMPONENTS OF STAPAC ON THE SEAT	LB
T1(3)	1	X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS OF STAPAC ON THE SEAT	FT-LB
TIN		TIME AT STAPAC INITIATION	SEC
ECA		SEAT TO GIMBAL ROLL EULER ANGLE AT THE CAGED POSITION	DEG

*These output quantities are states.

- STANDARD COMPONENT "SP" CALCULATES THE FORCES AND TORQUES THAT ACT ON THE SEAT FROM STAPAC

- THE EULER ANGLES FOR THE GYROSCOPE AND THE ROCKET ARE STATES WITH RESPECT TO THE SEAT COORDINATE SYSTEM

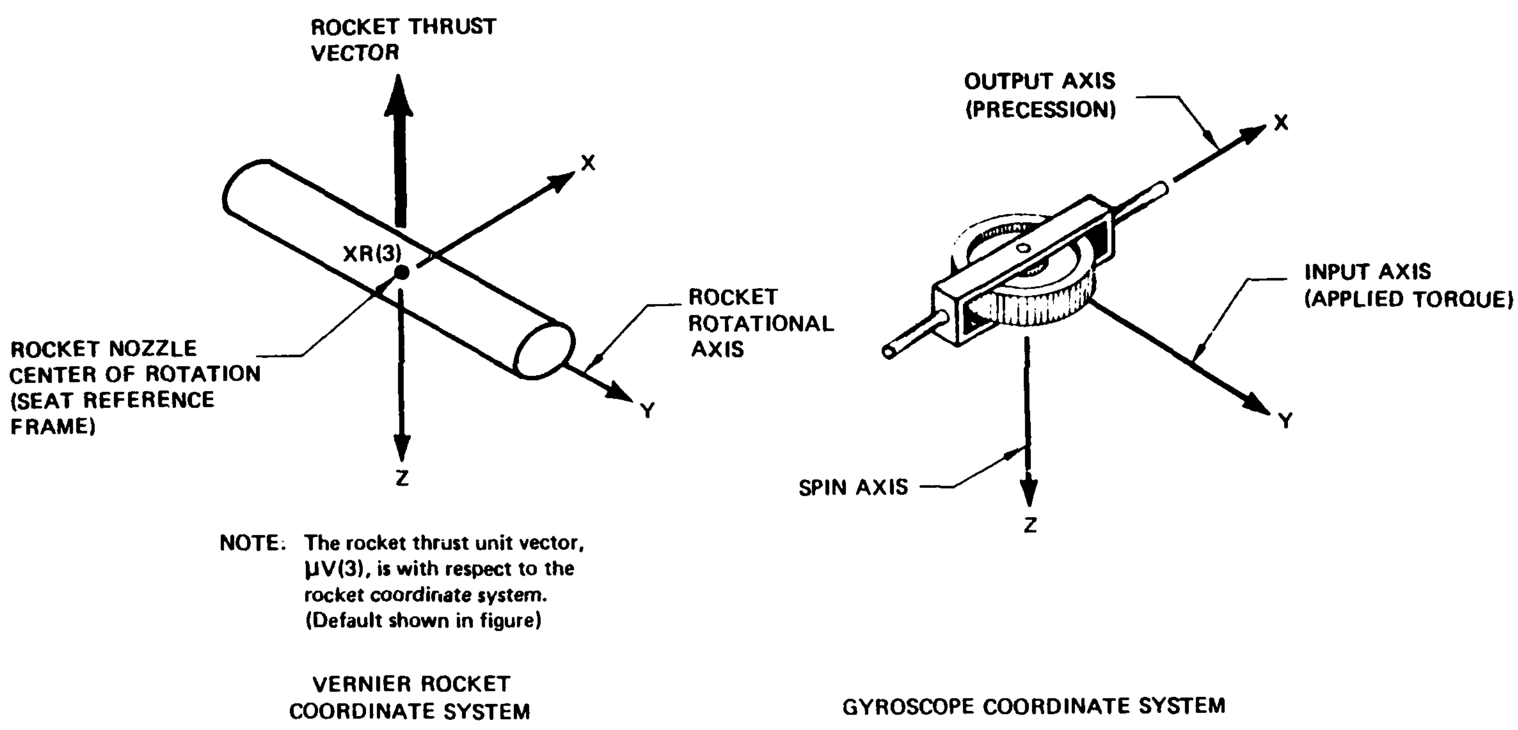


Figure 28. Standard Component "SP" Input/Output Overview

SP

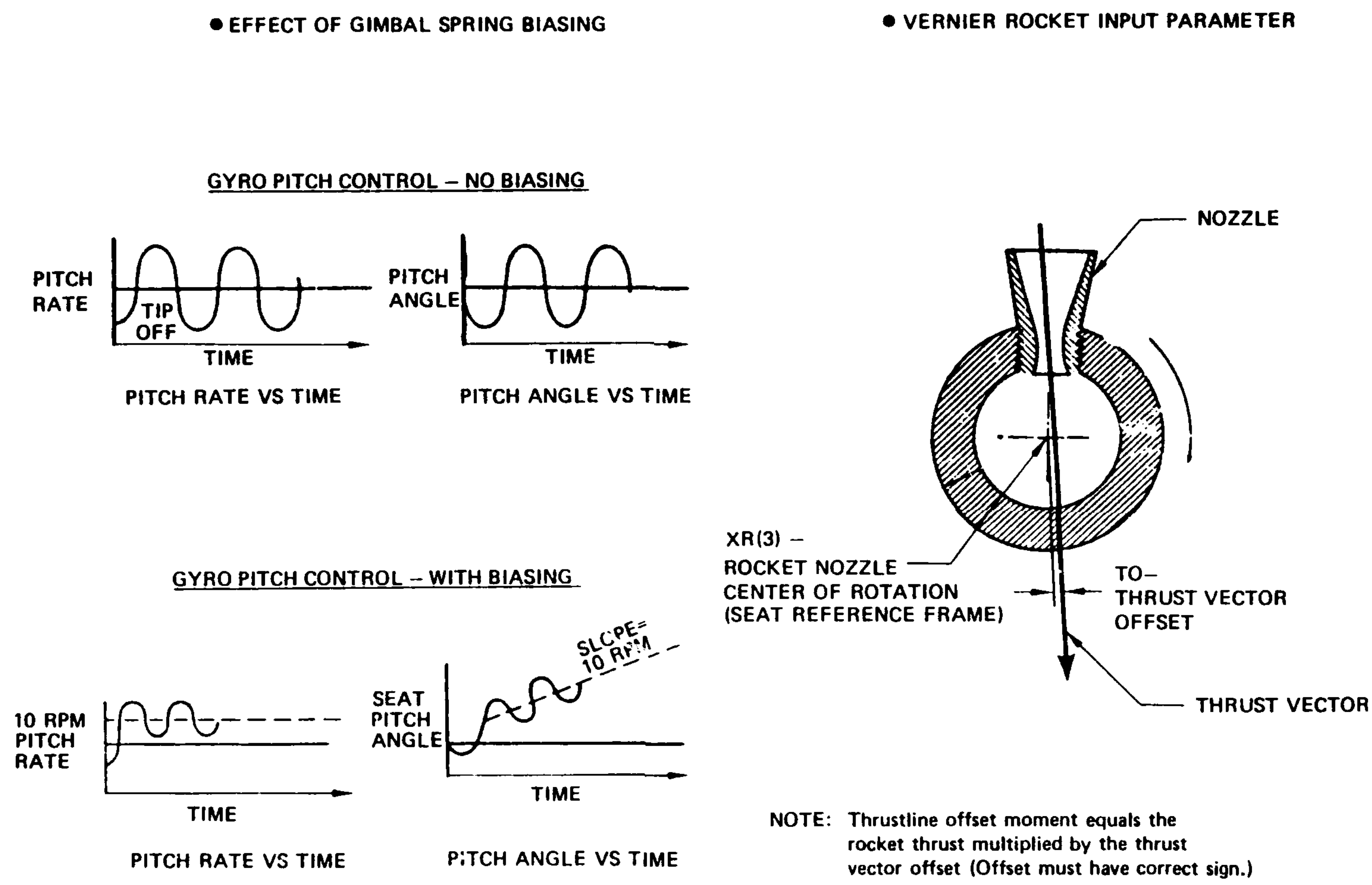


Figure 29. Standard Component "SP" Gimbal Spring and Vernier Rocket

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
TRF			ROCKET TABLE: TIME (INDEPENDENT FORCE (DEPENDENT)	 SEC LBS
FON		CT	SUSTAINER IGNITION FLAG (1 = ROCKET ON)	-
PCG(3)			X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE PROPELLANT CENTER OF GRAVITY	FT
EA(3)			SEAT TO ROCKET PROPELLANT EULER ANGLES (YAW,PITCH, ROLL)	DEG
XRN(3)			X,Y,Z PROPELLANT SYSTEM POSITION VECTOR OF THE ROCKET NOZZLE	FT
YAW			YAW EULER ANGLE OF THE THRUST VECTOR IN THE PROPELLANT COORDINATE SYSTEM	DEG
PIT			PITCH EULER ANGLE OF THE THRUST VECTOR IN THE PROPELLANT COORDINATE SYSTEM	DEG
PL			PROPELLANT GRAIN LENGTH	FT
POD			PROPELLANT GRAIN OUTSIDE DIAMETER	FT
PID			PROPELLANT GRAIN INSIDE DIAMETER	FT

NAME	PORT NO.	DESCRIPTION	UNITS
W*	1	WEIGHT OF UNBURNED PROPELLANT	LB
PHA		ROCKET PHASE 0 = BEFORE IGNITION 1 = ROCKET BURN 2 = ROCKET OFF	-
RON		ROCKET ON FLAG (1 = ON 0 = OFF)	
F1(3)	1	X,Y,Z SEAT BODY AXIS FORCE COMPONENTS	LB
T1(3)	1	X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS	FT-LB
X(3)	1	X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE PROPELLANT CENTER OF GRAVITY	FT
BM(3)	1	X,Y,Z UNBURNED ROCKET PROPELLANT MOMENTS OF INERTIA (IXX,IYY,IZZ)	SLUG-FT^2
BP(3)	1	UNBURNED ROCKET PROPELLANT PRODUCTS OF INERTIA (IXY, IXZ,IYZ)	SLUG-FT^2
FR		SUSTAINER ROCKET FORCE MAGNITUDE	LB
PWI		INITIAL WEIGHT OF THE PROPELLANT	LB
SPI		ROCKET PROPELLANT SPECIFIC IMPULSE	LB-SEC/LB
RHO		ROCKET PROPELLANT DENSITY	LB/FT^3
VWI		INITIAL VIRTUAL WEIGHT	LB
TMI(3)		PROPELLANT MOMENTS OF INERTIA AS IF IT WERE A SOLID GRAIN	SLUG-FT^2
TIG		ROCKET IGNITION TIME	SEC

**This output quantity is a state.*

NOTE: The yaw and pitch euler angles of the rocket nozzle are with respect to the propellant coordinate system. The thrust vector acts in the negative Z direction with respect to the nozzle coordinate system.

- STANDARD COMPONENT "SR" CALCULATES THE FORCES AND TORQUES THAT ACT ON THE SEAT FROM THE SUSTAINER ROCKET
- UPDATED INERTIAL PROPERTIES ARE FED TO STANDARD COMPONENT "WB" (WEIGHT AND BALANCE)

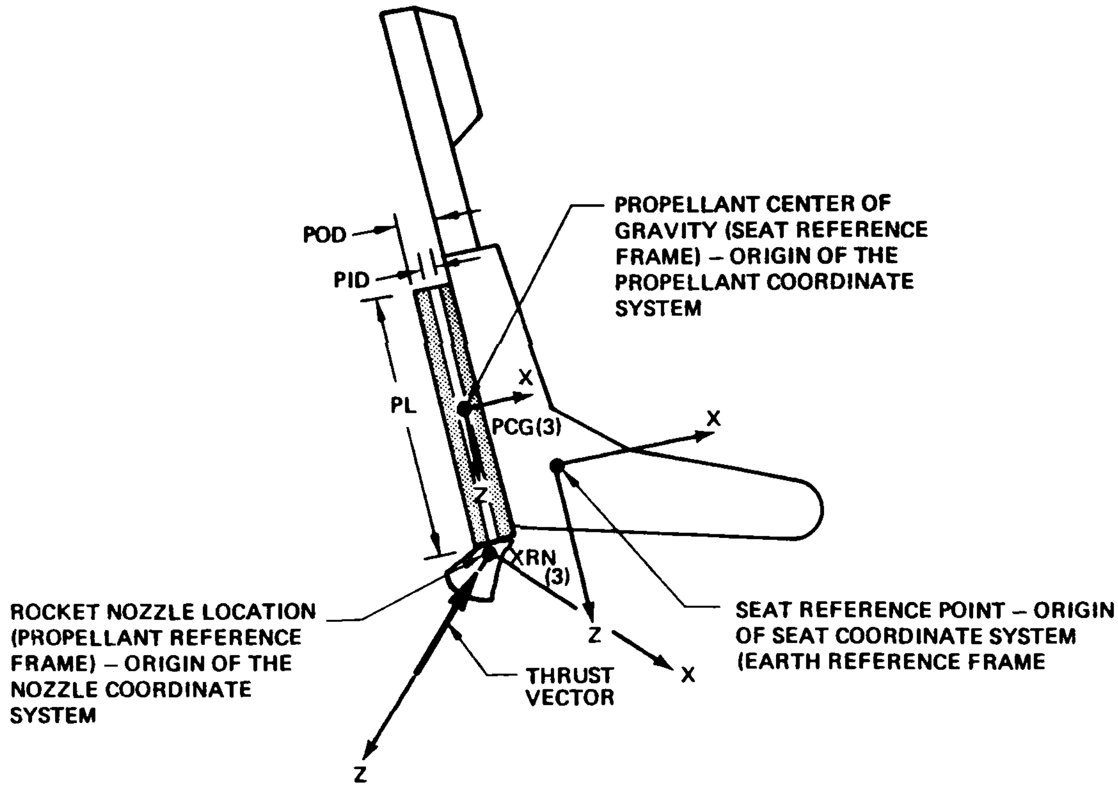


Figure 30. Standard Component "SR" Input/Output Overview

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
AB			NUMBER OF ATTACHED BODIES	-
SW			BASIC SEAT WEIGHT	LB
SX(3)			X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE BASIC SEAT CENTER OF GRAVITY	FT
SM(3)			MOMENT OF INERTIA VECTOR ABOUT THE C.G. FOR THE BASIC SEAT (IXX,IYY,IZZ)	SLUG-FT^2
SP(3)			PRODUCT OF INERTIA VECTOR ABOUT THE C.G. FOR THE BASIC SEAT (IXY,IXZ,IYZ)	SLUG-FT^2
W*	1	SR	WEIGHT OF BODY ONE	LB
X(3)*	1	SR	X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE CENTER OF GRAVITY FOR BODY ONE	FT
BM(3)*	1	SR	MOMENT OF INERTIA VECTOR FOR BODY ONE TRANSFORMED INTO THE SEAT SYSTEM (IXX,IYY,IZZ)	SLUG-FT^2
BP(3)*	1	SR	PRODUCT OF INERTIA VECTOR FOR BODY ONE TRANSFORMED INTO THE SEAT SYSTEM (IXY,IXZ,IYZ)	SLUG-FT2
W*	2		WEIGHT OF BODY TWO	LB
X(3)*	2		X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE CENTER OF GRAVITY FOR BODY TWO	FT
BM(3)*	2		MOMENT OF INERTIA VECTOR FOR BODY TWO TRANSFORMED INTO THE SEAT SYSTEM (IXX,IYY,IZZ)	SLUG-FT^2

*Default value = 0.

NAME	PORT NO.	NORMALLY DRIVEN BY	DESCRIPTION	UNITS
BP(3)*	2		PRODUCT OF INERTIA VECTOR FOR BODY TWO TRANSFORMED INTO THE SEAT SYSTEM (IXY,IXZ,IYZ)	SLUG-FT^2
W*	3		WEIGHT OF BODY THREE	LB
X(3)*	3		X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE CENTER OF GRAVITY FOR BODY THREE	FT
BMI(3)*	3		MOMENT OF INERTIA VECTOR FOR BODY THREE TRANSFORMED INTO THE SEAT SYSTEM (IXX,IYY,IZZ)	SLUG-FT^2
BP(3)*	3		PRODUCT OF INERTIA VECTOR FOR BODY THREE TRANSFORMED INTO THE SEAT SYSTEM (IXY,IXZ,IYZ)	SLUG-FT^2

*Default Value = 0.

Note - All moments and products of inertial must be rotated into the seat coordinate system.

NAME	PORT NO.	DESCRIPTION	UNITS
CW		COMPOSITE WEIGHT OF THE SEAT	LB
CCG(3)		X,Y,Z SEAT BODY AXIS COMPOSITE CENTER OF GRAVITY	FT
CMI(3)		MOMENT OF INERTIA VECTOR ABOUT THE C.G. FOR THE COMPOSITE SEAT (IXX,IYY,IZZ)	$SLUG\text{-}FT^2$
CPI(3)		PRODUCT OF INERTIA VECTOR ABOUT THE C.G. FOR THE COMPOSITE SEAT (IXY,IXZ,IYZ)	$SLUG\text{-}FT^2$

APPENDIX E

PROGRAM AEROMED

This appendix contains program AEROMED, the aeromedical post processor.

```
      PROGRAM AEROMED (OUTPUT,TAPE7,TAPE6=OUTPUT)
C
      DIMENSION TIME(4000),GX(4000),GY(4000),GZ(4000),DR(4000),
     .          RAD(4000),RADXY(4000),RADZ(4000)
C
C  NOTICE...........
C
C  A PORTION OF THIS PROGRAM EVALUATES ACCELERATION DATA
C  ESSENTIALLY IN ACCORDANCE WITH ACCEPTED AEROMEDICAL
C  PROCEDURES. THESE KINDS OF RESULTS ARE THEN NORMALLY USED
C  TOGETHER WITH OTHER FACTORS TO DETERMINE ACCEPTABILITY OF
C  ACCELERATION LOADS APPLIED TO THE EJECTEE.
C
C  THE EVALUATION METHOD HAS BEEN ADOPTED HERE TO SERVE AS A
C  FOUNDATION FOR CREATING FIGURES OF MERIT RELATED TO THE
C  PERFORMANCE OF A PARTICULAR ESCAPE SYSTEM CONFIGURATION.
C  THERE IS THEN AN OPPORTUNITY TO COMPARE, BY STANDARD MEANS,
C  THE PERFORMANCE OF ONE CONFIGURATION WITH THAT OF OTHERS
C  AND COMPARED IN TERMS OF THE MOST FUNDAMENTAL AND CRITICAL
C  PARAMETERS OF PERFORMANCE OF ANY ESCAPE SYSTEM, NAMELY,
C  ACCELERATION LOADS ON THE HUMAN EJECTEE.
C
C  THIS PROGRAM THEREFORE SERVES AS AN ENGINEERING TOOL ONLY
C  AND SHOULD NOT BE CONSIDERED AN ACCEPTABLE AEROMEDICAL
C  EVALUATION TOOL TO MEASURE ACCEPTABILITY OF AN ESCAPE SYSTEM
C  FOR SAFE OPERATIONAL USE.
C
C  **********  INITIALIZATION  **********
C
      GXMAX = GYMAX = GZMAX = DRMAX = RDMAX = -10.
      GXMIN = GYMIN = GZMIN = DRMIN = RDMIN =  10.
      TINJURY = EXPERNC = 0
C
C  *****  READ THE AEROMED PARAMETERS FROM TAPE7  *****
C
      REWIND 7
      READ(7,10) PRT,EXP,GXP,GXN,GYL,GZL,DRP,DRN,RDL
 10   FORMAT(9F12.4)
C
C  *****  READ THE AEROMED VARIABLES FROM TAPE7  *****
C
      NPTS = 4000
      I = 0
 20   I = I + 1
      IF(I.GT.4000) GO TO 35
      READ (7,10) TIME(I),DR(I),GX(I),GY(I),GZ(I)
C
      IF(EOF(7)) 30,25
C
 25   GX(I) = -GX(I)
      GY(I) = -GY(I)
      GO TO 20
C
 30   NPTS = I - 1
C
C  *****  CALCULATE RADXY AND RADZ  *****
C
 35   DO 40 I=1,NPTS
```

```
      GXL = GXP
      IF(GX(I).LT.0.0) GXL = GXN
      DRL = DRP
      IF(GX(I).LT.0.0) DRL = DRN
      RADXY(I) = (GX(I)/GXL)**2 + (GY(I)/GYL)**2
 40   RADZ(I) = (DR(I)/DRL)**2
C
C  *****  CALCULATE THE Z-AXIS TOLERANCE RATIO FOR EACH WINDOW  *****
C
      N = 0
 50   N = N + 1
      I = 0
      RADZMAX = 0
 60   J = N + I
C
      IF(TIME(J).GT.TIME(N)+0.063) GO TO 70
      IF(GZ(J).GE.0.) RADZMAX = AMAX1(RADZ(J),RADZMAX)
      IF(GZ(J).LT.0.) RADZMAX = AMAX1((GZ(J)/GZL)**2,RADZMAX)
      IF(J.EQ.NPTS) GO TO 90
      I = I + 1
      GO TO 60
C
C  DETERMINE THE ACCELERATION RADICAL ......
C
 70   RAD(N) = SQRT (RADZMAX + RADXY(N))
C
C  UPDATE THE UNSAFE LOAD EXPERIENCE FACTOR ......
C
      IF(RDL.GE.RAD(N)) GO TO 80
      TINJURY = TINJURY + (RAD(N)-RDL)**EXP * (TIME(N+1)-TIME(N))
C
C  UPDATE THE TOTAL LOAD EXPERIENCE FACTOR ......
C
 80   EXPERNC = EXPERNC + RAD(N)**EXP * (TIME(N+1)-TIME(N))
C
      GO TO 50
 90   N = N - 1
      TMAX = TIME(N)
C
C  *****  CALCULATE THE SAFE LOAD EXPERIENCE FACTOR  *****
C
      TINJURY = TINJURY/TIME(N)
      EXPERNC = EXPERNC/TIME(N)
      THREAT = EXPERNC - TINJURY
C
C  *****  CALCULATE THE MAXIMUM AND MINUMUM AEROMEDICAL VARIABLES  *****
C
      DO 100 I=1,N
C
      GXMAX = AMAX1(GXMAX,GX(I))
      IF(GXMAX.EQ.GX(I)) GXMAXT = TIME(I)
      GYMAX = AMAX1(GYMAX,GY(I))
      IF(GYMAX.EQ.GY(I)) GYMAXT = TIME(I)
      GZMAX = AMAX1(GZMAX,GZ(I))
      IF(GZMAX.EQ.GZ(I)) GZMAXT = TIME(I)
      DRMAX = AMAX1(DRMAX,DR(I))
      IF(DRMAX.EQ.DR(I)) DRMAXT = TIME(I)
      RDMAX = AMAX1(RDMAX,RAD(I))
```

```
      IF(RDMAX.EQ.RAD(I)) RDMAXT = TIME(I)
C
      GXMIN = AMIN1(GXMIN,GX(I))
      IF(GXMIN.EQ.GX(I)) GXMINT = TIME(I)
      GYMIN = AMIN1(GYMIN,GY(I))
      IF(GYMIN.EQ.GY(I)) GYMINT = TIME(I)
      GZMIN = AMIN1(GZMIN,GZ(I))
  100 IF(GZMIN.EQ.GZ(I)) GZMINT = TIME(I)
C
C  *****  WRITE TO THE OUTPUT FILE  *****
C
      WRITE (6,110) TMAX
  110 FORMAT (1H1, *HUMAN TOLERANCE ANALYSIS THROUGH *,F10.3,
     .         * SECONDS OF THE SIMULATION*///,
     .         * AEROMEDICAL SIGN CONVENTION ......*//,
     .         * GX = +ACCEL,  GY = +ACCEL, GZ = -ACCEL ......*///)
C
      IF(PRT.EQ.1.) WRITE(6,120)
  120 FORMAT (//4X,*TIME*,9X,*GX*,10X,*GY*,10X,*GZ*,10X,*DRI*,8X,*RAD*,//,
     .          1X,F7.3,4F12.2,F11.2)
C
      IF(PRT.EQ.1.) WRITE(6,130) (TIME(I),GX(I),GY(I),GZ(I),DR(I),
     .                            RAD(I),I=1,N)
  130 FORMAT(1X,F7.3,4F12.2,F11.2)
C
      WRITE(6,140) GXMAX,GXMAXT,GYMAX,GYMAXT,GZMAX,GZMAXT,
     .             GXMIN,GXMINT,GYMIN,GYMINT,GZMIN,GZMINT,
     .             DRMAX,DRMAXT,RDMAX,RDMAXT
  140 FORMAT (2(1H0/),* GXMAX = *,F14.2,*      TIME = *,F14.3,//,
     .                * GYMAX = *,F14.2,*      TIME = *,F14.3,//,
     .                * GZMAX = *,F14.2,*      TIME = *,F14.3,//,
     .                * GXMIN = *,F14.2,*      TIME = *,F14.3,//,
     .                * GYMIN = *,F14.2,*      TIME = *,F14.3,//,
     .                * GZMIN = *,F14.2,*      TIME = *,F14.3,//,
     .                * DRIMAX = *,F13.2,*      TIME = *,F14.3,//,
     .                * RADMAX = *,F13.2,*      TIME = *,F14.3)
C
      WRITE (6,150) EXPERNC,THREAT,TINJURY
  150 FORMAT (2(1H0/),* FIGURES OF MERIT.................*,////,
     .                * EXPERIENCE FACTOR -  TOTAL LOAD = *,
     .F14.3,//,
     .                * EXPERIENCE FACTOR -   SAFE LOAD = *,
     .F14.3,//,
     .                * EXPERIENCE FACTOR - UNSAFE LOAD = *,
     .F14.3)
C
      END
```

APPENDIX F

EASIEST PROCEDURE FILES

This appendix contains listings of the EASIEST procedure files. The procedure for attaching these files and submitting an EASIEST run is given in Section V.

```
EASIEST PROCEDURE FILE - LATEST REVISION DEC 12, 1980
*
THIS FILE CONTAINS THE CCL PROCEDURES REQUIRED TO EXECUTE AND MAINTAIN THE
EASIEST CREW ESCAPE SIMULATION PROGRAM
*
PROCEDURE DIRECTORY
*
     SUBRUN    -  PROCEDURE TO SUBMIT A BATCH EASIEST RUN
     DBFMOD    -  PROCEDURE TO MODIFY THE EASIEST DATA BASE FILE
     COMPILE   -  PROCEDURE TO COMPILE A SINGLE EASIEST COMPONENT
     COMPALL   -  PROCEDURE TO COMPILE AN ENTIRE SOURCE LIBRARY
     EZSTGEN   -  PROCEDURE TO GENERATE EASIEST FROM DELIVERY TAPE
*
SEE THE EASIEST MANUAL FOR COMPLETE USAGE INFORMATION
*
*EOR
.PROC,SUBRUN,MODFILE,ANLFILE,TIME=100,INOUT=100,
CORE=115000,IDENT=EZ5,COEF=0,NOLIST=OUTPUT/0,AEROMED=0/YES.
RETURN,JOB,PF,MODFILE,ANFILE.
REQUEST,JOB,*Q.
COPYCR,JOBFILE,JOB.
ATTACH,MODFILE.
COPYCF,MODFILE,JOB.
ATTACH,ANLFILE.
COPYCF,ANLFILE,JOB.
ROUTE(JOB,DC=IN,TID=Z1,ST=CSA)
RETURN,MODFILE,ANLFILE,JOB,JOBFILE.
.DATA,JOBFILE
IDENT,T  TIME,IO  INOUT,CM  CORE.    D790183,CREW ESCAPE EASIEST JOB
ATTACH(COMPLIB,MR=1)
ATTACH(EZSTLIB,MR=1)
LIBRARY(EZSTLIB,COMPLIB)
COPYCF,INPUT,MODEL.
REWIND,MODEL.
ATTACH(EASY5,MR=1)
ATTACH(TAPE78=EZSTDBF,MR=1)
MAP(OFF)
LDSET(PRESET=ZERO)
EASY5(MODEL)
RETURN(MODEL,EASY5,EASY,TAPE78,TAPE7,TAPE8,TAPE10,TAPE11,TAPE12)
REWIND(TAPE9)
RFL,CORE.
FTN(I=TAPE9,B=EZFORT,R=2,EL=F,L=NOLIST,ROUND)
COPYCF,INPUT,ANFIL.
REWIND,ANFIL.
RETURN(TAPE3)
IFE,.NOT.NUM(COEF),NOAIRP.
ATTACH(TAPE3=COEF,MR=1)
ENDIF,NOAIRP.
REWIND(EZFORT)
ATTACH(NONSIM5,MR=1)
COPYLM(NONSIM5,EZFORT,NONSIMT)
```

AD-A096 597 BOEING MILITARY AIRPLANE CO SEATTLE WA F/6 1/3
ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM. VOLUME --ETC(U)
SEP 80 C L WEST, B R UMMEL, R F YURCZYK F33615-79-C-3407
UNCLASSIFIED AFWAL-TR-80-3014-VOL-1 NL

4 OF 8

AD A
096597

```
REWIND(NONSIMT)
RETURN(EZFORT,NONSIM5,MAPFILE)
LDSET(PRESET=ZERO,MAP=SB/MAPFILE)
NONSIMT(ANFIL)
SKIP,NOMAP.
EXIT,U.
REWIND,MAPFILE.
COPYCF,MAPFILE,OUTPUT.
EXIT.
ENDIF,NOMAP.
IFE,.NOT.NUM(AEROMED),NOAERO
REWIND,TAPE7.
ATTACH(AROMEDB,MR=1)
LDSET(PRESET=ZERO)
AROMEDB.
RETURN,AROMEDB.
ENDIF,NOAERO.
EXIT,U.
REWIND(TAPE30)
RETURN(TAPE25,INIT,INTERP,NONSIM,SIBTCH,TFBTCH,RLBTCH)
RETURN(SMBTCH,ANFIL,NONSIMT)
ATTACH(NSMPPT,MR=1)
LDSET(PRESET=ZERO)
NSMPPT(PL=99999)
EXIT.
*EOR
*********************************************************************************
*EOR
.PROC,DBFMOD,INFILE,DBFILE=EZSTDBF,LSTFILE.
RETURN,TAPE3,TAPE78,COMPLIB,FILOAD5.
RETURN,LSTFILE,INFILE,PF,DBFILE,EZSTDBF.
ATTACH,TAPE3=INFILE.
EXIT,U.
ATTACH,TAPE78=DBFILE.
EXIT,U.
SET,R1=0.
IFE,FILE(TAPE78,AS),PURGE.
SET,R1=1.
ENDIF,PURGE.
REQUEST,TAPE79,*PF.
ATTACH,FILOAD5.
ATTACH,COMPLIB.
LIBRARY,COMPLIB.
MAP,OFF.
LDSET,PRESET=ZERO.
FILOAD5.
LIBRARY.
CATALOG,TAPE79,DBFILE,RP=999.
IFE,R1=1,NOPURGE.
PURGE,TAPE78.
ENDIF,NOPURGE.
RETURN,DBFILE.
CONNECT,OUTPUT.
COPYCR,MESFILE,OUTPUT.
```

```
RETURN,TAPE78,TAPE79,FILOAD5,COMPLIB,TAPE3,MESFILE.
IFE,FILE(TAPE9,AS),NOLIST.
REWIND,TAPE9.
COPYCF,TAPE9,LSTFILE.
RETURN,TAPE9.
COPYCR,MESFILE,OUTPUT.
ENDIF,NOLIST.
REVERT.
EXIT.
LIBRARY.
CONNECT,OUTPUT.
SKIPF,MESFILE,2.
COPYCR,MESFILE,OUTPUT.
RETURN,TAPE78,TAPE79,FILOAD5,COMPLIB,TAPE3,MESFILE.
REVERT.
.DATA,MESFILE.
DBFMOD PROCEDURE HAS SUCCESSFULLY EXECUTED.
A NEW CYCLE OF DBFILE HAS BEEN CREATED.
THE PREVIOUS HIGHEST NUMBERED CYCLE OF DBFILE
(IF ONE EXISTED) HAS BEEN PURGED.....
.EOR
COMPONENT INPUT DATA IS AVAILABLE ON LOCAL
FILE LSTFILE.....
.EOR
DBFMOD PROCEDURE HAS ABORTED......
NO NEW CYCLE OF DBFILE HAS BEEN CREATED...
PREVIOUS CYCLE (IF ANY) STILL EXISTS.
*EOR
*****************************************************************************
*EOR
.PROC,COMPILE,N,CODE=0.
RETURN,PF,EZSTFTN,ONEREL,ONEFTN,FTNLIST,LIBLIST.
REQUEST,FTNLIST,*Q.
ATTACH,EZSTFTN.
SKIPF,EZSTFTN,N.
BKSP,EZSTFTN.
COPYCR,EZSTFTN,ONEFTN.
REWIND,ONEFTN.
RETURN,EZSTFTN.
FTN,I=ONEFTN,B=ONEREL,R=2,L=FTNLIST.
ROUTE,FTNLIST,DC=PR,TID=Z1,ST=CSA,FID=F  N  CODE.
SKIP,A1.
EXIT,S.
REWIND,MESFILE.
CONNECT,OUTPUT.
COPYBR,MESFILE,OUTPUT.
RETURN,ONEFTN,ONEREL,MESFILE.
REVERT,ABORT.
ENDIF,A1.
RETURN,EZSTLIB.
ATTACH,EZSTLIB.
EDITLIB,I=DIRECT,L=LIBLIST.
EXTEND,EZSTLIB.
REWIND,MESFILE.
```

```
SKIPF,MESFILE.
CONNECT,OUTPUT.
COPYBR,MESFILE,OUTPUT.
RETURN,ONEFTN,EZSTLIB,ONEREL,LIBLIST,DIRECT,MESFILE.
REVERT.
EXIT,S.
REWIND,MESFILE.
SKIPF,MESFILE.
CONNECT,OUTPUT.
COPYBR,MESFILE,OUTPUT.
RETURN,DIRECT,ONEREL,MESFILE.
REVERT,ABORT.
.DATA,DIRECT.
LIBRARY(EZSTLIB,OLD)
REWIND(ONEREL)
REPLACE(*,ONEREL)
FINISH.
ENDRUN.
.EOF.
.DATA,MESFILE.
FORTRAN ERRORS DURING COMPILATION.PROCEDURE ABORTED
FORTRAN LISTING WITH ERROR DESCRIPTION AVAILABLE ON FILE FTNLIST
.EOR
COMPILE PROCEDURE SUCCESSFULLY EXECUTED
.EOR
LIBRARY MODIFY ERROR.....COMPILE PROCEDURE TERMINATED
.EOF
*EOR
********************************************************************************
*EOR
.PROC,COMPALL,SOURCE,LIBRARY,NOLIST.
RETURN,SOURCE,RELOC,PACK,LIBRARY.
ATTACH,SOURCE.
COMBINE,SOURCE,PACK,999.
RETURN,SOURCE.
REWIND,PACK.
FTN,I=PACK,L=LIST,B=RELOC,R=2,OPT=2,ROUND.
REQUEST,LIBARY,*PF.
EDITLIB,I=DIRECT,L=Ø.
CATALOG,LIBARY,RP=999.
EXIT,U.
RETURN,RELOC,PACK,LIBARY.
REVERT.
.DATA,DIRECT.
LIBRARY(LIBARY,NEW)
REWIND(RELOC)
ADD(*,RELOC)
FINISH.
ENDRUN.
.EOF
*EOR
********************************************************************************
```

```
*
PROCEDURE EZSTGEN
*
THIS PROCEDURE WILL GENERATE THE EASIEST PROGRAM FROM THE
EASIEST DELIVERY TAPE
*
INSTRUCTIONS
*
TO EXECUTE THIS PROCEDURE SUBMIT THE FOLOWING DECK TO
THE ASD COMPUTER INPUT QUEUE AFTER INSTRUCTING THE TAPE
LIBRARY TO MOUNT TAPE NUMBER L02377:
*
EZ5,T300,I01000,CM100000,NT1.          D79018383,EASIEST TAPE RUN
REQUEST,TAPE,NT,PE,VSN=L02377.
COPYBF,TAPE,TEMP.
BEGIN,EZSTGEN,TEMP,TPW=
*
SUBMITTING THE ABOVE DECK WILL BOOTSTRAP LOAD AND EXECUTE THE
FOLLOWING PROCEDURE
*
*EOR
.PROC,EZSTGEN,TPW.
REWIND,TAPE.
COPYTF,EZSTPRC,2
COPYTF,BACOMPS,1
COPYTF,COMPASS,1.
REQUEST,TSOUR,*PF.
COPYBR,BACOMPS,TSOUR,999.
COPYBR,COMPASS,TSOUR,999.
CATALOG,TSOUR.
RETURN,TSOUR.
RETURN,BACOMPS,COMPASS.
BEGIN,COMPALL,TEMP,TSOUR,COMPLIB,LIST=Ø.
ATTACH,TSOUR.
PURGE,TSOUR.
RETURN,TSOUR.
COPYTF,EZSTFTN,1.
BEGIN,COMPALL,TEMP,EZSTFTN,EZSTLIB,LIST=Ø.
COPYTF,FILOADS,1.
COMPL,FILOADS,FILOAD5,Ø.
COPYTF,FILDAT,1.
BEGIN,DBFMOD,TEMP,FILDAT.
COPYTF,EASYS,Ø.
COPYTF,EASY5,1
COMPL,EASYS,EASY5,1.
COPYTF,NONSIMS,Ø.
COPYTF,NONSIM5,1.
COMPL,NONSIMS,NONSIM5,1.
COPYTF,NSMPPTS,1.
COMPL,NSMPPTS,NSMPPT,Ø.
COPYTF,AEROMED,1.
COMPL,AEROMED,AROMEDB,Ø.
COPYTF,F4EMAN,2.
COPYTF,MCORR,2.
COPYTF,ACORR,2.
```

```
COPYTF,MODAPP,2.
COPYTF,ANALAPP,2.
BEGIN,SUBRUN,TEMP,MCORR,ACORR,TIME=1500,INOUT=1500,CORE=230000,AEROMED.
.DATA,COPYTF.
.PROC,COPYTF,FILE,CODE.
SET,R1=CODE.
RETURN,FILE.
REQUEST,FILE,*PF.
COPYBF,TAPE,FILE.
REWIND,FILE.
IFE,R1.NE.2,NOPASS.
IFE,R1.NE.1,NOPASS.
CATALOG,FILE,RP=999,TK=TPW.
REVERT.
ENDIF,NOPASS.
CATALOG,FILE,RP=999.
REWIND,FILE.
IFE,R1.NE.1,NODROP.
RETURN,FILE.
ENDIF,NODROP.
REVERT.
.EOF
.DATA,COMPL.
.PROC,COMPL,SOURCE,FILE,CODE.
SET,R1=CODE.
REWIND,SOURCE,PACK.
COMBINE,SOURCE,PACK,999.
RETURN,SOURCE,RELOC,PF.
REWIND,PACK.
REQUEST,PF,*PF.
IFE,R1,NE.Ø,NOCOPY.
FTN,I=PACK,L=Ø,ROUND,OPT=2,B=RELOC.
COPYLM,FILE,RELOC,PF.
RETURN,RELOC,FILE.
ELSE,NOCOPY.
FTN,I=PACK,L=Ø,OPT=2,ROUND,B=PF.
IFE,R1.NE.2,NOPASS.
ENDIF,NOCOPY.
CATALOG,PF,FILE,RP=999.
RETURN,PF,PACK.
REVERT.
ENDIF,NOPASS.
CATALOG,PF,FILE,RP=999.
RETURN,PF,PACK.
REVERT.
```

APPENDIX G

EASIEST STANDARD COMPONENTS

This appendix contains listings of the EASIEST standard components which include the following:

NAME	DESCRIPTION
AB	Attached body (Survival Kit)
AE	Airplane
AG	Atmospheric properties
AM	Aeromedical
AP	Aerodynamic plate
AS	Seat aerodynamics
CE	Crewperson
CS	Airplane control surfaces
CT	Catapult
DR	DART
GP	Simple parachute mortar and restraints
LI	Parachute lines
MP	Parachute mortar
PC	Parachute
RL	Rails
RS	Restraints
SE	Ejection seat
SL	Sled
SP	STAPAC
SR	Sustainer rocket
WB	Weight and balance

```
      SUBROUTINE AB (UAB,UABD,IUAB,XAB,XABD,IXAB,WAB,WABD,IWAB,
     .               EAB,EABD,IEAB,
     .               WT,BMI,BPI,FAB,TAB,FAU,TAU,TRM)
C
C         EASIEST ATTACHED BODY COMPONENT
C
C  DESIGNED BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C
C  ***************  AB OUTPUTS  ***************
C
C  LINEAR VELOCITIES - BODY AXIS
C
C     UAB(3)  - X,Y,Z LINEAR VELOCITY VECTOR OF THE ATTACHED
C               BODY (FT/SEC)
C     UABD(3) - X,Y,Z LINEAR VELOCITY RATE VECTOR OF THE ATTACHED
C               BODY (FT/SEC/SEC)
C     IUAB(3) - INTEGRATION CONTROL
C
C  LINEAR POSITIONS - EARTH SYSTEM
C
C     XAB(3)  - X,Y,Z LINEAR POSITION VECTOR OF THE ATTACHED BODY (FT)
C     XABD(3) - X,Y,Z LINEAR POSITION RATE VECTOR OF THE ATTACHED
C               BODY (FT/SEC)
C     IXAB(3) - INTEGRATION CONTROL
C
C  ANGULAR VELOCITIES - BODY AXIS
C
C     WAB(3)  - X,Y,Z ANGULAR VELOCITY COMPONENTS - P,Q,R  (DEG/SEC)
C     WABD(3) - X,Y,Z ANGULAR VELOCITY RATE COMPONENTS (DEG/SEC/SEC)
C     IWAB(3) - INTEGRATION CONTROL
C
C  EULER ANGLES  --  EARTH TO ATTACHED BODY  --  YAW,PITCH,ROLL
C
C     EAB(3)  - EARTH TO ATTACHED BODY EULER ANGLES (DEG)
C     EABD(3) - EULER ANGLE RATES (DEG/SEC)
C     IEAB(3) - INTEGRATION CONTORL
C
C  ***************  AB INPUTS  ***************
C
C     WT      - WEIGHT OF THE ATTACHED BODY (LB)
C     BMI(3) - ATTACHED BODY MOMENTS OF INERTIA - IXX,IYY,IZZ
C              (SLUG-FT**2)
C     BPI(3) - ATTACHED BODY PRODUCTS OF INERTIA - IXY,IXZ,IYZ
C              (SLUG-FT**2)
C     FAB(3) - X,Y,Z BODY AXIS FORCE COMPONENTS FROM THE RESTRAINTS (LB)
C     TAB(3) - X,Y,Z BODY AXIS TORQUE COMPONENTS FROM THE RESTRAINTS (LB)
C     FAU(3) - AUXILIARY X,Y,Z BODY AXIS FORCE COMPONENTS (LB)
C     TAU(3) - AUXILIARY X,Y,Z BODY AXIS TORQUE COMPONENTS (FT-LB)
C     TRM(3) - X,Y,Z PARENT BODY EARTH VELOCTIY COMPONENTS FOR
C              CALCULATING THE LINEAR POSITION RATES DURING TRIM (FT/SEC)
C
C  DIMENSIONS OF CALLING ARGUMENTS ......
C
      DIMENSION UAB(3),UABD(3),IUAB(3),XAB(3),XABD(3),IXAB(3),
     .          WAB(3),WABD(3),IWAB(3),EAB(3),EABD(3),IEAB(3),
     .          BMI(3),BPI(3),FAB(3),TAB(3),FAU(3),TAU(3),TRM(3)
```

```
C
C  INTERNAL DIMENSIONS ......
C
      DIMENSION TINER(3,3),TEMP1(3),TEMP2(3),TEMP3(3),WABIR(3),
     .          EABIR(3),DEA(3,3),DAE(3,3),F(3),T(3)
C
      COMMON /CICCAL/ ICCAL
      COMMON /COVRLY/ INST
      COMMON /CSSFLG/ SSFLG
      COMMON / CIO / IREAD,IWRITE,IDIAG
      DATA RPD,DPR / .01745329, 57.29578 /
      DATA GRAV /32.174/
C
C  ***************************
C  *****  INITIALIZATION  *****
C  ***************************
C
      IF(ICCAL.NE.1) GO TO 20
C
      DO 10 I=1,3
      IF(FAB(I) .EQ. 0.99999) FAB(I) = 0
      IF(FAU(I) .EQ. 0.99999) FAU(I) = 0
      IF(TAB(I) .EQ. 0.99999) TAB(I) = 0
 10   IF(TAU(I) .EQ. 0.99999) TAU(I) = 0
      TRM(1) = TRM(2) = TRM(3) = 0
C
C  ==================================================
C
C  SET UP THE ATTACHED BODY INERTIA TENSOR ......
C
 20   TINER(1,1) =  BMI(1)
      TINER(1,2) = -BPI(1)
      TINER(1,3) = -BPI(2)
      TINER(2,1) = -BPI(1)
      TINER(2,2) =  BMI(2)
      TINER(2,3) = -BPI(3)
      TINER(3,1) = -BPI(2)
      TINER(3,2) = -BPI(3)
      TINER(3,3) =  BMI(3)
C
C  CHANGE FROM DEGREES TO RADIANS ......
C
      DO 30 I=1,3
      WABIR(I) = WAB(I) * RPD
 30   EABIR(I) = EAB(I) * RPD
C
C  CALCULATE THE DIRECTION COSINE MATRICES ......
C
      CALL DIRCOS (DEA,EABIR)
      CALL TRANS (DAE,DEA,3,3)
C
C  CALCULATE THE TOTAL FORCE AND TORQUE DUE TO THE EXTERNAL
C  FORCES AND GRAVITY ......
C
      DO 40 I=1,3
      F(I) = FAB(I) + FAU(I) + WT * DEA(I,3) * SSFLG
 40   T(I) = TAB(I) + TAU(I)
C
```

```
C   ****************************************
C   *****  ANGULAR VELOCITY EQUATIONS  *****
C   ****************************************
C
C  CALCULATE TINER * WABIR ......
C
      CALL MATMPY (TEMP1,TINER,WABIR,3,3,1)
C
C  CALCULATE WABIR X (TINER * WABIR) ......
C
      CALL CRSPRD (TEMP2,WABIR,TEMP1)
C
C  SUM TERMS TO OBTAIN TOTAL TORQUE ......
C
      DO 50 I=1,3
 50   TEMP3(I) = T(I) - TEMP2(I)
C
C  CALCULATE WABDIR ......
C
      CALL LUEQS (TINER,TEMP1,TEMP3,TEMP2,3,1,3,3,3,1.E-14,IERROR)
      IF(IERROR.NE.1) GO TO 70
      WRITE(6,60)
 60   FORMAT(/* INERTIA MATRIX OF THE ATTACHED BODY IS SINGULAR ...*,
     .         *RUN STOPPED*/)
      STOP
C
 70   DO 80 I=1,3
 80   IF(IWAB(I).NE.0) WABD(I) = TEMP1(I) * DPR
C
C   **********************************
C   *****  EULER ANGLE EQUATIONS *****
C   **********************************
C
      CALL EARATE (TEMP1,WABIR,EABIR)
      DO 90 I=1,3
 90   IF(IEAB(I).NE.0) EABD(I) = TEMP1(I) * DPR
C
C   **************************************
C   *****  LINEAR VELOCITY EQUATIONS *****
C   **************************************
C
C  CALCULATE WABIR X UAB ......
C
      CALL CRSPRD (TEMP1,WABIR,UAB)
C
C  CALCULATE F/M ......
C
      ABMASS = WT/GRAV
      DO 100 I=1,3
 100  TEMP2(I) = F(I)/ABMASS
C
C  CALCULATE UABD ......
C
      DO 110 I=1,3
 110  IF(IUAB(I).NE.0) UABD(I) = TEMP2(I) - TEMP1(I)
C
C   **************************************
C   *****  LINEAR POSITION EQUATIONS *****
```

```
C   **************************************
C
      CALL MATMPY (TEMP1,DAE,DAB,3,3,1)
      DO 120 I=1,3
 120  IF(IXAB(I).NE.0) XABD(I) = TEMP1(I)
C
C  SUBTRACT TRIM VELOCITY FROM POSITION RATES DURING TRIM ......
C
      IF(INST.NE.31) GO TO 140
      DO 130 I=1,3
 130  IF(IXAB(I).NE.0) XABD(I) = XABD(I) - TRM(I)
C
 140  RETURN
      END
```

```
      SUBROUTINE AE (UAP,UAPD,IUAP,XAP,XAPD,IXAP,WAP,WAPD,IWAP,
     .                EAP,EAPD,IEAP,TRM,TRMD,ITRM,ALPHA,BETA,VMACH,ALT,
     .                AW,B,C,S,XCP,ZCP,AMI,API,
     .                XTHR,XAIL,XELE,XRUD,XEN,END,TALT,TVEL,
     .                FRA1,TRA1,FCA1,TCA1,FDA1,TDA1,
     .                FRA2,TRA2,FCA2,TCA2,FDA2,TDA2,CPF)
C
C                **** THE EASIEST AIRPLANE COMPONENT ****
C
C  THIS ROUTINE IS A SIX DEGREE OF FREEDOM MODEL OF AN AIRPLANE
C
C  THE AERODYNAMIC COEFFICEINTS ARE READ FROM TAPE3
C
C  THE AIRPLANE TRIM IS PROVIDED INTERNALLY USING EASY STEADY STATE
C  ANALYSIS
C
C  CONTROL SURFACE AND THRUST COMMANDS INPUT BY THE USER AFTER
C  TRIM WILL BE INTERPRETED AS BEING AN ADDITION TO THE SETTINGS
C  REQUIRED FOR TRIM
C
C
C  DESIGNED BY B. UMMEL AND C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C  ***************  AE OUTPUTS  ***************
C
C  LINEAR VELOCITIES - BODY AXIS
C
C     UAP(3)  - X,Y,Z LINEAR VELOCITY VECTOR OF THE AIRPLANE CENTER
C               OF GRAVITY (FT/SEC)
C     UAPD(3) - X,Y,Z LINEAR VELOCITY RATE VECTOR OF THE AIRPLANE
C               CENTER OF GRAVITY (FT/SEC/SEC)
C     IUAP(3) - INTEGRATION CONTROL
C
C  LINEAR POSITIONS - EARTH SYSTEM
C
C     XAP(3)  - X,Y,Z LINEAR POSITION VECTOR OF THE AIRPLANE CENTER
C               OF GRAVITY (FT)
C     XAPD(3) - X,Y,Z LINEAR POSITION RATE VECTOR OF THE AIRPLANE
C               CENTER OF GRAVITY (FT/SEC)
C     IXAP(3) - INTEGRATION CONTROL
C
C  ANGULAR VELOCITIES - BODY AXIS
C
C     WAP(3)  - X,Y,Z ANGULAR VELOCITY COMPONENTS - P,Q,R (DEG/SEC)
C     WAPD(3) - X,Y,Z ANGULAR VELOCITY RATE COMPONENTS (DEG/SEC/SEC)
C     IWAP(3) - INTEGRATION CONTROL
C
C  EULER ANGLES  --  EARTH TO BODY AXIS  --  YAW,PITCH,ROLL
C
C     EAP(3)  - EARTH TO AIRPLANE EULER ANGLES (DEG)
C     EAPD(3) - EULER ANGLE RATES (DEG/SEC)
C     IEAP(3) - INTEGRATION CONTROL
C
C  TRIM CONTROL STATES  --  TRM(4),TRMD(4),ITRM(4)
C
C     TRM(1) = TRIM THROTTLE SETTING
C     TRM(2) = TRIM AILERON SETTING
```

```
C     TRM(3) = TRIM ELEVATOR SETTING
C     TRM(4) = TRIM RUDDER SETTING
C
C     ALPHA - ANGLE OF ATTACK (DEG)
C     BETA  - SIDESLIP ANGLE (DEG)
C     VMACH - MACH NUMBER
C     ALT   -  ALTITUDE ABOVE SEA LEVEL (FT)
C
C  ***************  AE INPUTS  ***************
C
C          AW       - AIRPLANE WEIGHT (LB)
C          B        - WINGSPAN (FEET)
C          C        - MEAN AERODYNAMIC CHORD (FEET)
C          S        - REFERENCE AREA (FT**2)
C          XCP      - AIRPLANE BODY X-AXIS POSITON OF THE CENTER
C                     OF PRESSURE (FT)
C          ZCP      - AIRPLANE BODY Z-AXIS POSITION OF THE CENTER
C                     OF PRESSURE (FT)
C          AMI(3)   - MOMENTS OF INERTIA -- IXX,IYY,IZZ
C                     (SLUG-FT**2)
C          API(3)   - PRODUCTS OF INERTIA -- IXY,IXZ,IYZ
C                     (SLUG-FT**2)
C          XTHR     - EXTERNAL THRUST SETTING (LB)
C          XAIL     - EXTERNAL AILERON SETTING (DEG)
C          XELE     - EXTERNAL ELEVATOR SETTING (DEG)
C          XRUD     - EXTERNAL RUDDER SETTING (DEG)
C          XEN(3)   - X,Y,Z AIRPLANE BODY AXIS POSITION VECTOR
C                     OF THE ENGINE (FT)
C          END(3)   - AIRPLANE BODY AXIS DIRECTION COSINES
C                     OF THE ENGINE THRUST VECTOR
C          TALT     - DESIRED TRIM AIRPLANE ALTITUDE (FT)
C          TVEL     - DESIRED TRIM AIRPLANE SPEED (FT/SEC)
C          FRA1(3) - PORT ONE X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS
C                     ACTING ON THE AIRPLANE FROM THE RAILS (LB)
C          TRA1(3) - PORT ONE X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS
C                     ACTING ON THE AIRPLANE FROM THE RAILS (FT-LB)
C          FCA1(3) - PORT ONE X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS
C                     ACTING ON THE AIRPLANE FROM THE CATAPULT (LB)
C          TCA1(3) - PORT ONE X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS
C                     ACTING ON THE AIRPLANE FROM THE CATAPULT (FT-LB)
C          FDA1(3) - PORT ONE X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS
C                     ACTING ON THE AIRPLANE FROM THE DART (LB)
C          TDA1(3) - PORT ONE X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS
C                     ACTING ON THE AIRPLANE FROM THE DART (FT-LB)
C          FRA2(3) - PORT TWO X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS
C                   - ACTING ON THE AIRPLANE FROM THE RAILS (LB)
C          TRA2(3) - PORT TWO X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS
C                     ACTING ON THE AIRPLANE FROM THE RAILS (FT-LB)
C          FCA2(3) - PORT TWO X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS
C                     ACTING ON THE AIRPLANE FROM THE CATAPULT (LB)
C          TCA2(3) - PORT TWO X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS
C                     ACTING ON THE AIRPLANE FROM THE CATAPULT (FT-LB)
C          FDA2(3) - PORT TWO X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS
C                     ACTING ON THE AIRPLANE FROM THE DART (LB)
C          TDA2(3) - PORT TWO X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS
C                     ACTING ON THE AIRPLANE FROM THE DART (FT-LB)
C          CPF      - PRINT FLAG FOR AERODYNAMIC COEFFICIENTS
C
```

```
C   ***************  DATA DECLARATIONS ***************
C
C
C   CALLING SEQUENCE DIMENSIONS ......
C
      DIMENSION XAP(3),XAPD(3),IXAP(3),UAP(3),UAPD(3),IUAP(3),
     .          WAP(3),WAPD(3),IWAP(3),EAP(3),EAPD(3),IEAP(3),
     .          TRM(4),TRMD(4),ITRM(4),
     .          AMI(3),API(3),XEN(3),END(3),
     .          FRA1(3),TRA1(3),FCA1(3),TCA1(3),FDA1(3),TDA1(3),
     .          FRA2(3),TRA2(3),FCA2(3),TCA2(3),FDA2(3),TDA2(3)
C
C   INTERNAL DIMENSIONS ......
C
      DIMENSION UW(3),UG(3),UWB(3),DEA(3,3),DAE(3,3),TINER(3,3),
     .          TEMPIN(3,3),F(3),FGRAV(3),FENG(3),FCOR(3),FRCD(3),
     .          FAERO(3),T(3),TENG(3),TRCD(3),TAERO(3),WAPIRS(3),
     .          TEMP1(3),TEMP2(3),TEMP3(3),EAPIR(3),WAPIR(3),
     .          R(2000),ITPLOT(60),FORMAT(8),TITLE(6)
C
      COMMON/REGIONS/NR(60)
      COMMON/CICCAL/ICCAL
      COMMON/COVRLY/INST
      COMMON/CIO/ IREAD,IWRITE,IDIAG
      DATA GRAV /32.174/
C
C
      DATA RPD,DPR,IFLAG / .01745329, 57.29578, 0 /
C
C   ***************************
C   *****  INITIALIZATION  *****
C   ***************************
C
      IF(ICCAL .NE. 1) GO TO 110
      XAP(3) = -TALT
      UAP(1) =  TVEL
      IF(XCP .EQ. 0.99999) XCP = 0
      IF(ZCP .EQ. 0.99999) ZCP = 0
      IF(XTHR .EQ. 0.99999) XTHR = 0
      IF(XAIL .EQ. 0.99999) XAIL = 0
      IF(XELE .EQ. 0.99999) XELE = 0
      IF(XRUD .EQ. 0.99999) XRUD = 0
      IF(CPF .EQ. 0.99999) CPF = 0
C
      DO 5 I=1,3
      IF(FRA1(I) .EQ. 0.99999) FRA1(I) = 0
      IF(TRA1(I) .EQ. 0.99999) TRA1(I) = 0
      IF(FCA1(I) .EQ. 0.99999) FCA1(I) = 0
      IF(TCA1(I) .EQ. 0.99999) TCA1(I) = 0
      IF(FDA1(I) .EQ. 0.99999) FDA1(I) = 0
      IF(TDA1(I) .EQ. 0.99999) TDA1(I) = 0
      IF(FRA2(I) .EQ. 0.99999) FRA2(I) = 0
      IF(TRA2(I) .EQ. 0.99999) TRA2(I) = 0
      IF(FCA2(I) .EQ. 0.99999) FCA2(I) = 0
      IF(TCA2(I) .EQ. 0.99999) TCA2(I) = 0
      IF(FDA2(I) .EQ. 0.99999) FDA2(I) = 0
      IF(TDA2(I) .EQ. 0.99999) TDA2(I) = 0
  5   CONTINUE
```

```
C
C  -----  SET UP AIRPLANE INERTIA TENSOR  -----
C
      TINER(1,1) =  AMI(1)
      TINER(1,2) = -API(1)
      TINER(1,3) = -API(2)
      TINER(2,1) = -API(1)
      TINER(2,2) =  AMI(2)
      TINER(2,3) - -API(3)
      TINER(3,1) = -API(2)
      TINER(3,2) = -API(3)
      TINER(3,3) =  AMI(3)
C
C  ------  READ AERODYNAMIC TABLES FROM TAPE3  ------
C
      IF(IFLAG.EQ.1) GO TO 110
      IFLAG=1
      REWIND 3
C
      IF(CPF.EQ.1.0)WRITE(6,10)
 10   FORMAT(25X,*AERODYNAMIC COEFFICIENTS FOR THE AIRPLANE*)
      READ(3,20) NTEMP
 20   FORMAT(18I4)
      READ(3,20) (ITPLOT(I),I=1,NTEMP)
      NTEPM1=NTEMP-1
      NR(1)=5
      DO 30 I=1,NTEPM1
 30   NR(I+1)=NR(I)+ITPLOT(I)+1
      IF(CPF.EQ.1.0)WRITE(6,40) (I,NR(I),I=1,NTEMP)
 40   FORMAT(9(I3,I5))
      IF(CPF.EQ.1.0)WRITE(6,50)
 50   FORMAT(1H0)
C
 60   READ(3,70) (ITPLOT(I),I=1,4),(TITLE(I),I=1,6)
 70   FORMAT(4I4,6A10)
      NTEMP=ITPLOT(1)
      IF(NTEMP.LE.0) GO TO 110
      NTEMP=NR(NTEMP)+ITPLOT(2)
      NTEMP1=ITPLOT(3)
      NTEMP1=NR(NTEMP1)+ITPLOT(4)
      READ(3,80) FORMAT
 80   FORMAT(6A10)
      READ(3,FORMAT) (R(I),I=NTEMP,NTEMP1)
      IF(CPF.EQ.1.0)WRITE(6,90)(ITPLOT(I),I=1,4),(TITLE(I),I=1,6)
 90   FORMAT(4I6,6A10)
      IF(CPF.EQ.1.0)WRITE(6,100)(I,R(I),I=NTEMP,NTEMP1)
 100  FORMAT(4(I6,F14.6))
      GO TO 60
C
C  ////////////////////////////////////////////////////////////////////
C
C  ----- CONVERT ANGULAR RATES AND EULER ANGLES TO RADIANS -----
C
 110  DO 120 I=1,3
      EAPIR(I) = EAP(I)*RPD
 120  WAPIR(I) = WAP(I)*RPD
C
C  ----- COMPUTE EARTH TO AIRPLANE AND AIRPLANE TO EARTH  ----
```

```
C                       DIRECTION COSINE MATRICES
C
      CALL DIRCOS (DEA,EAPIR)
      CALL TRANS (DAE,DEA,3,3)
C
C  ------  CONTROL SURFACE SETTINGS  ------
C
      DA=2.*(TRM(2)+XAIL)
      ADA = ABS(DA)
      DE=(TRM(3)+XELE)
      DR=(TRM(4)+XRUD)
C
C  ------ OBTAIN SPEED OF SOUND, AIR DENSITY, AND WIND VELOCITY ------
C
      ALT = -XAP(3)
      CALL ATMOS (AZ,RHO,ALT,UW,0,0,0)
C
C  ------  PUT WIND INTO BODY COORDINATES  ------
C
      CALL MATMPY (UWB,DEA,UW,3,3,1)
C
C  ------  ADD WIND VELOCITY TO AIRPLANE VELOCITY  ------
C
      U0(1)=UAP(1)-UWB(1)
      U0(2)=UAP(2)-UWB(2)
      U0(3)=UAP(3)-UWB(3)
C
C  ------  AERO VARIABLES  ------
C
      IF(U0(1).EQ.0.0.AND.U0(3).EQ.0.0)U0(1)=.01
      ALPHA = ARTAN2(U0(3),U0(1))*DPR
      COSA = COS(ALPHA*RPD)
      SINA = SIN(ALPHA*RPD)
      CALL DOTPRD (VBAR2,U0,U0,3)
      VBAR= SQRT(VBAR2)
      BETA= ASIN(U0(2)/VBAR)*DPR
      VMACH = VBAR/AZ
      QAS = .5*RHO*VBAR2*S
C
C ----- COMPUTE STABILTY AXIS ANGULAR RATES -----
C
      WAPIRS(1) = WAPIR(1)*COSA + WAPIR(3)*SINA
      WAPIRS(2) = WAPIR(2)
      WAPIRS(3) = -WAPIR(1)*SINA + WAPIR(3)*COSA
C
C  ****************************************************************
C  **********  CALCULATE THE AERODYNAMIC COEFFICIENTS  *************
C  ****************************************************************
C
C  ------  TRANSFER AERO VARIABLES TO THE R ARRAY  ------
C
      R(1) = VMACH
      R(2) = ALPHA
      R(3) = BETA
C
C ------ Z AXIS FORCE COEFFICIENTS
C
C  BIAS COEFFICIENT FOR TRIM ......
```

```
      CALL LOOK (NR(1),R,CZO)
C  VARIATION OF CZO WITH ALPHA DOT ......
      CALL LOOK (NR(2),R,CZAD)
C  VARIATION OF CZO WITH PITCH RATE ......
      CALL LOOK (NR(3),R,CZQ)
C  VARIATION OF CZO WITH ELEVATOR POSITION ......
      CALL LOOK (NR(4),R,CZDE)
C  VARIATION OF CZO WITH AILERON POSITION ......
      CALL LOOK (NR(5),R,CZDA)
C
C ------- X-AXIS FORCE COEFFICIENTS
C
C  BIAS COEFFICIENT FOR TRIM ......
      R(4)=CZO
      CALL LOOK (NR(6),R,CXO)
C  VARIATION OF CXO WITH AILERON POSITION ......
      CALL LOOK (NR(7),R,CXDA)
C
C ------ PITCHING MOMENT COEFFICIENTS
C
C  BIAS COEFFICIENT FOR TRIM ......
      CALL LOOK (NR(8),R,CMO)
C  VARIATION OF CMO WITH ALPHA DOT ......
      CALL LOOK (NR(9),R,CMAD)
C  VARIATION OF CMO WITH PITCH RATE ......
      CALL LOOK (NR(10),R,CMQ)
C  VARIATION OF CMO WITH ELEVATOR POSITION ......
      CALL LOOK (NR(11),R,CMDE)
C  VARIATION OF CMO WITH AILERON POSITION ......
      CALL LOOK (NR(12),R,CMDA)
C
C ------ SIDE FORCE COEFFICIENTS
C
C  VARIATION OF CY WITH BETA ......
      CALL LOOK (NR(13),R,CYB)
C  VARIATION OF CY WITH ROLL RATE ......
      CALL LOOK (NR(14),R,CYP)
C  VARIATION OF CY WITH YAW RATE ......
      CALL LOOK (NR(15),R,CYR)
C  VARIATION OF CY WITH RUDDER POSITION ......
      CALL LOOK (NR(16),R,CYDR)
C  VARIATION OF CY WITH AILERON DEFLECTION ......
      CALL LOOK (NR(17),R,CYDA)
C
C ------ ROLLING MOMENT COEFFICIENTS
C
C  VARIATION OF CL WITH BETA ......
      CALL LOOK (NR(18),R,CLB)
C  VARIATION OF CL WITH ROLL RATE ......
      CALL LOOK (NR(19),R,CLP)
C  VARIATION OF CL WITH YAW RATE ......
      CALL LOOK (NR(20),R,CLR)
C  VARIATION OF CL WITH RUDDER DEFLECTION ......
      CALL LOOK (NR(21),R,CLDR)
C  VARIATION OF CL WITH AILERON DEFLECTION ......
      CALL LOOK (NR(22),R,CLDA)
C
C ------ YAWING MOMENT COEFFICIENTS
```

```
C
C  VARIATION OF CN WITH BETA ......
      CALL LOOK (NR(23),R,CNB)
C  VARIATION OF CN WITH ROLL RATE ......
      CALL LOOK (NR(24),R,CNP)
C  VARIATION OF CN WITH YAW RATE ......
      CALL LOOK (NR(25),R,CNR)
C  VARIATION OF CN WITH RUDDER DEFLECTION ......
      CALL LOOK (NR(26),R,CNDR)
C  VARIATION OF CN WITH AILERON DEFLECTION ......
      CALL LOOK (NR(27),R,CNDA)
C
C ----- PRINT AERO COEFFICIENTS DURING PRINT TASK ONLY
C
      IF(INST.EQ.60)WRITE(6,125)VMACH,ALPHA,BETA,CZO,CZAD,
     . CZQ,CZDE,CZDA,CXO,CXDA,CMO,CMAD,CMQ,CMDE,CMDA,CYB,CYP,
     . CYR,CYDR,CYDA,CLB,CLP,CLR,CLDR,CLDA,CNB,CNP,CNR,CNDR,CNDA
 125  FORMAT(/* AIRPLANE AERO COEFFICIENTS FOR MACH=*,G12.5,
     . * ALPHA=*,G12.5,* BETA=*,G12.5/* CZO =*,G12.5,* CZAD=*,G12.5,
     . * CZQ =*,G12.5,* CZDE=*,G12.5,* CZDA=*,G12.5,* CXO =*,G12.5/
     . * CXDA=*,G12.5,* CMO =*,G12.5,* CMAD=*,G12.5,* CMQ =*,G12.5,
     . * CMDE=*,G12.5,* CMDA=*,G12.5/* CYB =*,G12.5,* CYP =*,G12.5,
     . * CYR =*,G12.5,* CYDR=*,G12.5,* CYDA=*,G12.5,* CLB =*,G12.5/
     . * CLP =*,G12.5,* CLR =*,G12.5,* CLDR=*,G12.5,* CLDA=*,G12.5,
     . * CNB =*,G12.5,* CNP =*,G12.5/* CNR =*,G12.5,* CNDR=*,G12.5,
     . * CNDA=*,G12.5)
C
C  **************************************
C  *****  LINEAR VELOCITY EQUATIONS  *****
C  **************************************
C
C  ------  COMPUTE THE FORCE DUE TO GRAVITY  ------
C
      AMASS = AW/GRAV
      FGRAV(1) = AW * DEA(1,3)
      FGRAV(2) = AW * DEA(2,3)
      FGRAV(3) = AW * DEA(3,3)
C
C  ------  CALCULATE THE FORCE DUE TO THE CORIOLIS ACCELERATION  ------
C
      CALL CRSPRD (FCOR,WAPIR,UAP)
      DO 130 I=1,3
 130  FCOR(I) = -FCOR(I) * AMASS
C
C  ------  CALCULATE THE ENGINE FORCES  ------
C
      ETHRUST = TRM(1)+XTHR
      DO 140 I=1,3
 140  FENG(I) = ETHRUST * END(I)
C
C  ------  COMPUTE THE FORCES FROM THE  ------
C         RAILS, CATAPULTS, AND THE DARTS
C
      DO 150 I=1,3
 150  FRCD(I) = FRA1(I) + FRA2(I) + FCA1(I) + FCA2(I) +
     .             FDA1(I) + FDA2(I)
C
C  ------  CALCULATE THE BODY AXIS AERODYNAMIC FORCES  ------
```

```
C            (EXCEPT THOSE USING ALPHA DOT)
C
      BO2V = B/(VBAR+VBAR)
      CO2V = C/(VBAR+VBAR)
C
      FX = QAS*(CXO+CXDA*ADA)
      FY=QAS*(CYB*BETA+(CYP*WAPIRS(1)+CYR*WAPIRS(3))*BO2V+CYDR*DR
     .   +CYDA*DA)
      FZ = QAS*(CZO+CZDE*DE+CZDA*ADA+CO2V*CZQ*WAPIRS(2))
C
C  CHANGE FROM STABILITY AXIS TO BODY AXIS ......
C
      FAERO(1) = FZ * SINA - FX * COSA
      FAERO(2) = FY
      FAERO(3) = -FZ * COSA - FX * SINA
C
C --- TOTAL FORCES ACTING ON AIRPLANE EXCEPT FOR ALPHA DOT EFFECTS
C
      DO 160 I=1,3
 160  F(I) = FGRAV(I) + FCOR(I) + FENG(I) + FRCD(I) + FAERO(I)
C
C  --- SOLVE FOR LINEAR ACCELERATIONS USING FORCES INCLUDING ALPHA DOT EFFECT
C
      VAR = (CO2V * CZAD * QAS  ) / (UAP(1)**2 + UAP(3)**2)
      DEN = AMASS - VAR * (UAP(1)*COSA - UAP(3)*SINA)
C
      TEMP1(1) = (F(1)-(F(1)*COSA + F(3)*SINA)*VAR*UAP(1)/AMASS)/DEN
      TEMP1(2) = F(2) / AMASS
      TEMP1(3) = (F(3)-(F(1)*COSA + F(3)*SINA)*VAR*UAP(3)/AMASS)/DEN
C
      DO 170 I=1,3
 170  IF(IUAP(I).NE.0) UAPD(I) = TEMP1(I)
C
C  ************************************
C  *****  LINEAR POSITION EQUATIONS  *****
C  ************************************
C
      CALL MATMPY (TEMP1,DAE,UAP,3,3,1)
      DO 180 I=1,3
 180  IF(IXAP(I).NE.0) XAPD(I) = TEMP1(I)
C
C  **************************************
C  *****  ANGULAR VELOCITY EQUATIONS  *****
C  **************************************
C
C  ------  CALCULATE THE ENGINE TORQUE  ------
C
      CALL CRSPRD (TENG,XEN,FENG)
C
C  ------  CALCULATE THE TORQUE DUE TO THE  ------
C              RAILS, CATAPULTS, AND DARTS
C
      DO 190 I=1,3
 190  TRCD(I) = TRA1(I) + TRA2(I) + TCA1(I) + TCA2(I) +
     .            TDA1(I) + TDA2(I)
C
C  ------  CALCULATE THE AERODYNAMIC TORQUE  ------
C
```

```
      ALDOT = (UAP(1)*UAPD(3)-UAP(3)*UAPD(1))/(UAP(3)**2+UAP(1)**2)
C
      TX=QAS*B*(CLB*BETA+(CLP*WAPIRS(1)+CLR*WAPIRS(3))*BO2V+CLDR*DR
     .         +CLDA*DA)
      TY=QAS*C*(CMO+CO2V*(CMAD*ALDOT+CMQ*WAPIRS(2))+CMDE*DE+CMDA*ADA)
      TZ=QAS*B*(CNB*BETA+(CNP*WAPIRS(1)+CNR*WAPIRS(3))*BO2V+CNDR*DR
     .         + CNDA*DA)
C
C  CHANGE FROM STABILITY AXIS TO BODY AXIS ......
C
      TAERO(1) = TX * COSA - TZ * SINA - ZCP * FAERO(2)
      TAERO(2) = TY - XCP * FAERO(3) + ZCP * FAERO(1)
      TAERO(3) = TX * SINA + TZ * COSA + XCP * FAERO(2)
C
C  ------  CALCULATE THE TOTAL TORQUE ACTING ON THE AIRPLANE  ------
C
      DO 200 I=1,3
 200  T(I) = TENG(I) + TRCD(I) + TAERO(I)
C
C ---- PRINT AIRPLANE FORCES AND TORQUES DURING PRINT TASK ONLY
C
      IF(INST.EQ.60)WRITE(6,210)(FGRAV(I),I=1,3),(FCOR(I),I=1,3),
     .  (FENG(I),I=1,3),(TENG(I),I=1,3),(FRCD(I),I=1,3),(TRCD(I),I=1,3),
     .  (FAERO(I),I=1,3),(TAERO(I),I=1,3),FX,FY,FZ,TX,TY,TZ,ALDOT
 210  FORMAT(* AIRPLANE FORCES AND TORQUES*/* FRC.GRAV.    =*,3G12.5,
     .  * FRC.COREOLIS=*,3G12.5/* FRC.ENGINE  =*,3G12.5,* TRQ.ENGINE  =*
     . ,3G12.5/* FRC.EJSEAT  =*,3G12.5,* TRQ.EJSEAT  =*,3G12.5/
     . * FRC.AERO     =*,3G12.5,* TRQ.AERO     =*,3G12.5/
     . * FAERO.ST.AX.=*,3G12.5,* TAERO.ST.AX.=*,3G12.5/
     . * AIRPLANE ALPHA DOT=*,G12.5//)
C
C  CALCULATE TINER * WAPIR ......
C
      CALL MATMPY (TEMP1,TINER,WAPIR,3,3,1)
C
C  CALCULATE WAPIR X (TINER * WAPIR)
C
      CALL CRSPRD (TEMP2,WAPIR,TEMP1)
C
C  SUM TERMS TO OBTAIN TOTAL TORQUE ......
C
      DO 220 I=1,3
 220  TEMP3(I) = T(I) - TEMP2(I)
C
C  SET UP TEMPORARY INERTIA TENSOR ......
C
      DO 230 I=1,3
      DO 230 J=1,3
 230  TEMPIN(I,J) = TINER(I,J)
C
C  CALCULATE WAPD ......
C
      CALL LEQS (TEMPIN,TEMP1,TEMP3,TEMP2,3,1,3,3,3,1.E-14,IERROR)
      IF(IERROR.NE.1) GO TO 250
      WRITE(6,240)
 240  FORMAT(* INERTIA MATRIX OF AIRPLANE IS SINGULAR...RUN STOPPED*)
      STOP
 250  CONTINUE
```

```
C
      DO 260 I=1,3
 260  IF(IWAP(I).NE.0) WAPD(I) = TEMP1(I)*DPR
C
C   ***********************************
C   *****  EULER ANGLE EQUATIONS  *****
C   ***********************************
C
      CALL EARATE (TEMP1,WAPIR,EAPIR)
      DO 270 I=1,3
 270  IF(IEAP(I).NE.0) EAPD(I) = TEMP1(I)*DPR
C
C   *********************************
C   **********  TRIM LOGIC  **********
C   *********************************
C
      TRMD(1)=TRMD(2)=TRMD(3)=TRMD(4)=0
      IF(INST.NE.31) GO TO 280
      IF(ITRM(1).NE.0) TRMD(1) = TVEL - VBAR
      IF(ITRM(2).NE.0) TRMD(2) = + .01*WAPIR(1)+ EAPIR(3)
      IF(ITRM(3).NE.0) TRMD(3) = +.01*WAPIR(2)-.001*XAPD(3)
     .                           -.0001*(TALT+XAP(3))
      IF(ITRM(4).NE.0) TRMD(4) = +.01*WAPIR(3)
C
 280  RETURN
      END
```

```
      SUBROUTINE AG (VS,RHO,
     .                    H,WIN,BP,TE,SW)
      DIMENSION WIN(3),WIND(3)
      COMMON /CICCAL/ ICCAL
      COMMON /CSSFLG/ SSFLG
      COMMON /COVRLY/ INST
      COMMON /CIO/ IREAD,IWRITE,IDIAG
      DATA FL1,FL2 /0,0/
C
C  DESIGNED BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C  THE STANDARD COMPONENT WHICH DETERMINES THE AIR DENSITY, SPEED OF
C  SOUND, AND THE WIND VELOCITY AT A PRESCRIBED ALTITUDE IN A STANDARD
C  OR NON-STANDARD ATMOSPHERE.  IN ADDITION, IT SETS A FLAG WHICH FORCES
C  THE ACCELERATION OF GRAVITY TO BE ZERO FOR THE STEADY STATE CALCULATION
C  OF AN UNSUPPORTED SEAT.  THIS FLAG CAN ALSO BE USED TO ASSIST THE STEADY
C  STATE SOLVER WITH A SUPPORTED SEAT, AS EXPLAINED IN THE DOCUMENT.
C  THIS COMPONENT MUST BE INCLUDED IN THE MODEL GENERATION PROGRAM INPUT
C  FILE FOR ALL EASIEST MODELS.
C
C  ***************  AG OUTPUTS  ***************
C
C      VS  - VELOCITY OF SOUND (FT/SEC)
C      RHO - AIR DENSITY (SLUG/FT**3)
C
C  ***************  AG INPUTS  ***************
C
C      H   - HEIGHT ABOVE SEA LEVEL
C      WIN - X,Y,Z EARTH SYSTEM WIND COMPONENTS (FT/SEC)
C      BP  - BAROMETRIC PRESSURE AT THE REFERENCE ALTITUDE (IN. HG)
C              (AN UNINITIALIZED OR NON-POSITINVE VALUE OF BP
C               CAUSES A STANDARD ATMOSPHERE TO BE USED)
C      TE  - TEMPERATURE AT THE REFERENCE ALTITUDE (DEG F)
C      SW  - GRAVITY SWITCH FOR UNSUPPORTED SEAT STEADY STATE CALCULATION
C                    0 = GRAVITY OFF  (UNSUPPORTED SEAT)
C                    1 = GRAVITY ON
C
C  ////////////////////////////////////////////////////////////////
C
      DO 5 I=1,3
 5    WIND(I) = WIN(I)
C
      SSFLG = 1.
      IF(SW.NE.0) FL1 = SW
      IF(FL1.EQ.0 .AND. INST.EQ.31) SSFLG = 0
C
C  *****  CHECK TO SEE IF THE CALC XIC COMMAND HAS BEEN GIVEN  *****
C
      IF(FL2.EQ.1.) GO TO 70
      IF(ICCAL.EQ.1) GO TO 20
      WRITE(6,10)
 10   FORMAT(//5X,*WARNING -  THE CALC XIC COMMAND HAS NOT BEEN*,
     .        * GIVEN ...... EXECUTION TERMINATED. *,//)
      STOP
C
C  ***************************
C  *****  INITIALIZATION  *****
```

```
C   ****************************
C
 20     VS = RHO = 0
        IF(SW.EQ.0.99999) SW = 1.0
        FL1 = SW
        FL2 = 1.
        IF(BP .LE. 0.0 .OR. BP .EQ. .99999) BP=0.0
        BPE = BP
        IF(BPE .EQ. 0.0) H = TE = 0.0
        IF(BPE .EQ. 0.0) GO TO 70
        ABP = (BP * 144.)/2.036
        ATE = TE + 460.
        TO = ATE + 0.003566 * H
        TRATIO = ATE/TO
        PO = ABP * (TRATIO)**5.256
        GO TO 70
C
        ENTRY ATMOS
C
        DO 30 I=1,3
 30     WIN(I) = WIND(I)
C
        IF(BPE .NE. 0.0) GO TO 60
C
C  *****  STANDARD ATMOSPHERE  *****
C
        IF(H.GT.35332.) GO TO 40
C
C  ALTITUDE BELOW THE TROPOPAUSE ......
C
        TRATIO = 1.0 - 0.0000066709 * H
        PRATIO = TRATIO**5.256
        VS = 1116.75 * SQRT(TRATIO)
        GO TO 50
C
C  ALTITUDE ABOVE THE TROPOPAUSE ......
C
 40     PRATIO = 10.**((4705.-H)/48211.)
        VS = 970.9579
C
 50     RHO = 2962.*PRATIO/(VS**2)
        GO TO 70
C
C  *****  NON-STANDARD ATMOSPHERE  *****
C
 60     T = TO - 0.003566 * H
        P = PO * (T/TO)**5.256
        VS = (49.02) * SQRT(T)
        RHO = P/(1715.*T)
C
 70     RETURN
        END
```

```
      SUBROUTINE AM (DRE,RAD,PTS,PTI,
     .                FL,PRT,EXP,GXP,GXN,GYL,GZL,DRP,DRN,RDL,
     .                DR,GX,GY,GZ)
C
C  THIS ROUTINE WRITES ONTO TAPE7 AEROMEDICAL PARAMETERS AND VARIABLES
C  TO BE USED BY PROGRAM AEROMED.  NO MORE THAN 4000 VARIABLE SETS ARE
C  WRITTEN AT A TIME INTERVAL OF NO LESS THAN 0.001 SECONDS.
C
C  DESIGN BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C  *****  OUTPUTS  *****
C
C      DRE - DYNAMIC RESPONSE
C      RAD - ACCELERATION RADICAL
C      PTS - CURRENT NUMBER OF DATA SETS WRITTEN TO TAPE3
C      PTI - VALUE OF TIME WHEN THE LAST DATA SET WAS WRITTEN
C            ONTO TAPE3
C
C  *****  INPUTS  *****
C
C      FL  - FLAG TO INITIATE AEROMED CALCULATION (1 = START)
C      PRT - PROGRAM AEROMED FLAG TO PRINT THE LOAD FACTORS, DYNAMIC
C            RESPONSE, AND THE ACCELERATION RADICAL (1 = PRINT)
C      EXP - MEDICAL INJURY EXPONENT
C      GXP - THE LIMIT VALUE FOR THE X-AXIS POSITIVE AEROMED
C            LOAD FACTOR (G)
C      GXN - THE LIMIT VALUE FOR THE X-AXIS NEGATIVE AEROMED
C            LOAD FACTOR (G)
C      GYL - THE LIMIT VALUE FOR THE Y-AXIS AEROMED LOAD FACTOR (G)
C      GZL - THE LIMIT VALUE FOR THE Z-AXIS NEGATIVE AEROMED LOAD
C            FACTOR (G)
C      DRP - LIMIT VALUE OF THE DYNAMIC RESPONSE WHEN THE ACCELERATION
C            VECTOR IS FORWARD OF THE PLANE OF THE SEAT BACK
C      DRN - LIMIT VALUE OF THE DYNAMIC RESPONSE WHEN THE ACCELERATION
C            VECTOR IS AFT OF THE PLANE OF THE SEAT BACK
C      RDL - ACCELERATION RADICAL LIMIT
C      DR  - DYNAMIC RESPONSE
C      GX  - X AXIS LOAD FACTOR (G)
C      GY  - Y AXIS LOAD FACTOR (G)
C      GZ  - Z AXIS LOAD FACTOR (G)
C
      COMMON /CICCAL/ ICCAL
      COMMON /COVRLY/ INST
      COMMON /CTIME/ TIME
      COMMON /CPFLAG/ DUM,ITINC
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
C  ***************************
C  *****  INITIALIZATION *****
C  ***************************
C
      IF(ICCAL.NE.1) GO TO 20
      IF(PRT.EQ.0.99999) PRT = 0.
      IF(EXP.EQ.0.99999) EXP = 2.
      IF(GXP.EQ.0.99999) GXP = 35.
      IF(GXN.EQ.0.99999) GXN = 30.
      IF(GYL.EQ.0.99999) GYL = 15.
```

```
      IF(GZL.EQ.0.99999) GZL = 12.
      IF(DRP.EQ.0.99999) DRP = 18.
      IF(DRN.EQ.0.99999) DRN = 16.
      IF(RDL.EQ.0.99999) RDL = 1.0
      PTS = 0
      PTI = 0
C
C  WRITE AEROMEDICAL PARAMETERS ONTO TAPE7 ......
C
      WRITE(7,10) PRT,EXP,GXP,GXN,GYL,GZL,DRP,DRN,RDL
 10   FORMAT(9F12.4)
C
C  ////////////////////////////////////////////////////////////
C
 20   DRE = DR
C
C  CALCULATE THE ACCELERATION RADICAL ......
C
      GXL = GXP
      IF(-GX .LT. 0) GXL = GXN
      DRL = DRP
      IF(-GX .LT. 0) DRL = DRN
      IF(GZ .GE. 0) RADZ = (DR/DRL)**2
      IF(GZ .LT. 0) RADZ = (GZ/GZL)**2
      RAD = SQRT((GX/GXL)**2 + (GY/GYL)**2 + RADZ)
C
C  WRITE AEROMEDICAL VARIABLES ONTO TAPE7 ......
C
      IF(FL.NE.1.) GO TO 30
      IF(PTS.GE.4000.) GO TO 30
      IF(TIME.LT.PTI+.001) GO TO 30
      IF(ITINC.NE.1) GO TO 30
C
      IF(INST.EQ.26) WRITE (7,10) TIME,DR,GX,GY,GZ
      PTI = TIME
      PTS = PTS + 1.
C
 30   RETURN
      END
```

```
      SUBROUTINE AP (TCX,TCZ,
     .               F,T,SW,ALPHA,CX,CZ,
     .               UP,XPC,PA,EPL,ZEM,SRP,UST,EST,WST,XAP,EAP)
C
C ***** FORCES AND MOMENTS ON A SEAT FROM AN ATTACHED PLATE *****
C
C DESIGNED BY C.L. WEST
C LAST MODIFIED -  DECEMBER 6, 1980
C
C ***************  AP TABLES  ***************
C
C     TCX - PLATE SYSTEM X-AXIS FORCE COEFFICIENT TABLE
C
C              THE INDEPENDENT VARIABLE IS THE PLATE ANGLE OF ATTACK (DEG).
C              THE DEPENDENT VARIABLE IS THE PLATE X-AXIS FORCE COEFFICIENT.
C
C     TCZ - PLATE SYSTEM Z-AXIS FORCE COEFFICIENT TABLE
C
C              THE INDEPENDENT VARIABLE IS THE PLATE ANGLE OF ATTACK (DEG).
C              THE DEPENDENT VARIABLE IS THE PLATE Z-AXIS FORCE COEFFICIENT.
C
C ***************  AP OUTPUTS  ***************
C
C     F(3) - X,Y,Z SEAT BODY AXIS FORCE COMPONENTS (LB)
C     T(3) - X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS (FT-LB)
C     SW   - FLAG SET WHEN THE PLATE CENTROID PENETRATES THE
C            WINDSTREAM (1 = PENETRATION)
C     ALPHA - PLATE ANGLE OF ATTACK (DEG)
C     CX    - X AXIS FORCE COEFFICIENT
C     CZ    - Z AXIS FORCE COEFFICIENT
C
C ***************  AP INPUTS  ***************
C
C     UP        - EJECTION DIRECTION FLAG WRT THE AIRPLANE
C                 (1 = UPWARD   -1 = DOWNWARD)
C     XPC(3)    - X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE
C                 PLATE CENTROID (FT)
C     PA        - REFERENCE AREA OF THE ATTACHED PLATE (FT**2)
C     EPL(3)    - SEAT TO PLATE EULER ANGLES (DEG)
C     ZEM       - AIRPLANE BODY Z-AXIS POSITION OF THE PLATE CENTROID
C                 WHEN IT ENTERS THE WINDSTREAM (FT)
C                 --  SET TO ZERO WHEN INITIALLY IN WINDSTREAM  --
C     SRP(3)    - X,Y,Z EARTH SYSTEM POSITION VECTOR OF THE SEAT
C                 REFERENCE POINT (FT)
C     UST(3)    - X,Y,Z SEAT BODY AXIS SYSTEM VELOCITY COMPONENTS
C                 OF THE SEAT (FT/SEC)
C     EST(3)    - EARTH TO SEAT EULER ANGLES (DEG)
C     WST(3)    - X,Y,Z SEAT BODY AXIS SYSTEM ANGULAR VELOCITY
C                 COMPONENTS OF THE SEAT (DEG/SEC)
C     XAP(3)    - X,Y,Z EARTH SYSTEM POSITION VECTOR OF THE
C                 AIRPLANE CENTER OF GRAVITY (FT)
C     EAP(3)    - EARTH TO AIRPLANE EULER ANGLES (DEG)
C
C ///////////////////////////////////////////////////////////////////////
C
C CALLING SEQUENCE DIMENSIONS ......
C
      DIMENSION TCX(5),TCZ(5),F(3),T(3),XPC(3),EPL(3),
```

```
     .           SRP(3),UST(3),EST(3),WST(3),XAP(3),EAP(3)
C
C  INTERNAL DIMENSIONS ......
C
      DIMENSION EPLIR(3),EAPIR(3),ESTIR(3),WSTIR(3),DES(3,3),
     .           DEST(3,3),DEA(3,3),XPCA(3),XPCE(3),UPLE(3),
     .           DSP(3,3),DEP(3,3),UPL(3),UW(3),UO(3),
     .           DPS(3,3),FP(3)
C
      COMMON /CTIME/ TIME
      COMMON /CICCAL/ ICCAL
      COMMON /COVRLY/ INST
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
      DATA FP(2) / 0. /
      DATA RPD,DPR / .01745329, 57.29578 /
C
C  ****************************
C  *****  INITIALIZATION  *****
C  ****************************
C
      IF(ICCAL.NE.1) GO TO 20
      IF(UP.EQ.0.99999) UP = 1.
      IF(ZEM.EQ.0.99999) ZEM = 0
      SW = 0
      IF(ZEM.EQ.0) SW = 1.
      DO 10 I=1,3
      IF(XAP(I) .EQ. 0.99999) XAP(I) = 0
      IF(EAP(I) .EQ. 0.99999) EAP(I) = 0
      F(I) = 0
 10   T(I) = 0
C
C  ////////////////////////////////////////////////////////////////
C
C  BYPASS ROUTINE IF DURING STEADY STATE WHEN THE PLATE IS NOT INITIALLY
C  IN THE WINDSTREAM ......
C
 20   IF(INST.EQ.31 .AND. SW.EQ.0) GO TO 70
C
C  CONVERT FROM DEGREES TO RADIANS ......
C
      DO 30 I=1,3
 30   ESTIR(I) = EST(I) * RPD
C
C  CALCULATE THE DIRECTION COSINE MATRICIES ......
C
      CALL DIRCOS (DES,ESTIR)
      CALL TRANS (DEST,DES,3,3)
C
C  CONTROL FLAGS ......
C
      IF(SW.EQ.1.0) GO TO 50
C
C  CALCULATE THE CENTROID POSITION IN THE AIRPLANE SYSTEM ......
C
      DO 40 I=1,3
 40   EAPIR(I) = EAP(I) * RPD
      CALL DIRCOS (DEA,EAPIR)
```

```
      CALL VECXYZ (XPCE,XPC,SRP,DEST,2)
      CALL VECXYZ (XPCA,XPCE,XAP,DEA,1)
C
C  RETURN IF THE PLATE HAS NOT PENETRATED THE WINDSTREAM ......
C
      IF(ZEM*UP.LT.XPCA(3)*UP) GO TO 70
C
C  WRITE EMERGENCE MESSAGE ......
C
      IF(INST.EQ.26 .AND. SW.EQ.0) WRITE(6,45) TIME
 45   FORMAT(/5X,*AERODYNAMIC PLATE PENETRATION AT TIME = *,
     .         F10.4,* SEC*/)
      IF(ICCAL.NE.1) SW = 1.
C
C  *****  PLATE PENETRATION  *****
C
C  CONVERT FROM DEGREES TO RADIANS ......
C
 50   DO 55 I=1,3
      EPLIR(I) = EPL(I) * RPD
 55   WSTIR(I) = WST(I) * RPD
C
C  CALCULATE THE DIRECTION COSINE MATRICIES ......
C
      CALL DIRCOS (DSP,EPLIR)
      CALL TRANS (DPS,DSP,3,3)
      CALL MATMPY (DEP,DSP,DES,3,3,3)
C
C  DETERMINE THE VELOCITY OF THE PLATE CENTROID IN THE EARTH
C  SYSTEM  ......
C
      CALL VELXYZ (UPLE,UST,XPC,WSTIR,DEST)
C
C  OBTAIN THE AIR DENSITY AND WIND VELOCITY ......
C
      CALL ATMOS (VS,RHO,-SRP(3),UW,0,0,0)
C
C  SUBTRACT THE WIND VELOCITY FROM THE PLATE VELOCITY ......
C
      DO 60 I=1,3
 60   UO(I) = UPLE(I) - UW(I)
C
C  TRANSFORM THE EARTH VELOCITY INTO THE PLATE SYSTEM ......
C
      CALL MATMPY (UPL,DEP,UO,3,3,1)
C
C  CALCULATE THE AIRSPEED OF THE PLATE ......
C
      CALL DOTPRD (VBAR2,UPL,UPL,3)
C
C  DETERMINE THE PLATE ANGLE OF ATTACK ......
C
      ALPHA = ARTAN2(UPL(3),UPL(1)) * DPR
C
C  PERFORM THE TABLE SEARCH FOR CX AND CALCLATE ITS FORCE ......
C
      NTCX = TCX(2)
      CX = TBLU1 (ALPHA,TCX(4),TCX(NTCX+4),1,-NTCX)
```

```
      FP(1) = CX * .5 * RHO * VBAR2 * PA
C
C  PERFORM THE TABLE SEARCH FOR CZ AND CALCULATE ITS FORCE ......
C
      NTCZ = TCZ(2)
      CZ = TBLU1 (ALPHA,TCZ(4),TCZ(NTCZ+4),1,-NTCZ)
      FP(3) = CZ * .5 * RHO * VBAR2 * PA
C
C  TRANSFORM THE FORCES TO THE SEAT SYSTEM ......
C
      CALL TRANS (DPS,DSP,3,3)
      CALL MATMPY (F,DPS,FP,3,3,1)
C
C  CALCULATE THE MOMENTS ON THE SEAT FROM THE PLATE ......
C
      CALL CRSPRD (T,XPC,F)
C
 70   RETURN
      END
```

```
      SUBROUTINE AS (TAE,
     .               F,T,ALPHA,BETA,VMACH,Q,CX,CY,CZ,CL,CM,CN,EXL,EXA,
     .               CENT,TCZ,HD,
     .               OFF,UP,ZWS,XEM,CDX,ECX,ECY,ECZ,CLP,CMQ,CNR,S,SRP,
     .               UST,EST,WST,DSA,SRA,RON)
C
C  ***************  AS TABLES  ***************
C
C     TAE - EXPOSED AREA TABLE
C
C           THE INDEPENDENT VARIABLE IS THE EXPOSED LENGTH (FT).
C           THE DEPENDENT VARIABLE IS THE EXPOSED AREA (FT**2)
C
C  ***************  AS OUTPUTS  ***************
C
C     F(3)     - X,Y,Z SEAT BODY AXIS AERODYNAMIC FORCE COMPONENTS (LB)
C     T(3)     - X,Y,Z SEAT BODY AXIS AERODYNAMIC TORQUE COMPONENTS (FT-LB)
C     ALPHA    - SEAT ANGLE OF ATTACK (DEG)
C     BETA     - SEAT SIDESLIP ANGLE (DEG)
C     VMACH    - SEAT MACH NUMBER
C     Q        - DYNAMIC PRESSURE (LB/FT**2)
C     CX       - SEAT BODY X-AXIS FORCE COEFFICIENT
C     CY       - SEAT BODY Y-AXIS FORCE COEFFICIENT
C     CZ       - SEAT BODY A-AXIS FORCE COEFFICIENT
C     CL       - SEAT BODY AXIS ROLLING MOMENT COEFFICIENT
C     CM       - SEAT BODY AXIS PITCHING MOMENT COEFFICIENT
C     CN       - SEAT BODY AXIS YAWING MOMENT COEFFICIENT
C     EXL      - SEAT/CREWP EXPOSED LENGTH DURING EMERGENCE (FT)
C     EXA      - SEAT/CREWP EXPOSED AREA DURING EMERGENCE (FT**2)
C     CENT(3) - X,Y,Z SEAT BODY AXIS POSITON VECTOR OF THE
C               CENTROID OF THE EMERGED AREA (FT)
C     TCZ(20) - SEAT CENTROID LOCATION ARRAY (FT)
C     HD       - HYDRAULIC DIAMETER (FT)
C
C  ***************  AS INPUTS  ***************
C
C     OFF - FLAG TO INDICATE SEAT/RAIL SEPARATION (1 = SEPARATION)
C     UP  - EJECTION DIRECTION FLAG WRT THE AIRPLANE
C           (+1 = UPWARD    -1 = DOWNWARD)
C     ZWS - AIRPLANE BODY Z-AXIS POSITION OF THE WINDSTREAM
C           BOUNDRY LAYER AT THE POINT OF SEAT PENETRATION (FT)
C     XEM - X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE INITIAL
C           POINT TO PENETRATE THE WINDSTREAM (FT)
C     CDX - SEAT BODY X-AXIS POSITON OF THE CENTER OF PRESSURE
C           DURING SEAT EMERGENCE (FT)
C     ECX - SEAT BODY X-AXIS EMERGENCE COEFFICIENT
C     ECY - SEAT BODY Y-AXIS EMERGENCE COEFFICIENT
C     ECZ - SEAT BODY Z-AXIS EMERGENCE COEFFICIENT
C     CLP - ROLL DAMPING DERIVATIVE (DEG-1)
C     CMQ - PITCH DAMPING DERIVATIVE (DEG-1)
C     CNR - YAW DAMPING DERIVATIVE (DEG-1)
C     S      - SEAT REFERENCE AREA (FT**2)
C     SRP(3) - X,Y,Z EARTH POSITON VECTOR OF THE SEAT REFERENCE POINT (FT)
C     UST(3) - X,Y,Z SEAT BODY AXIS SYSTEM VELOCITY COMPONENTS
C              OF THE SEAT (FT/SEC)
C     EST(3) - EARTH TO SEAT EULER ANGLES (DEG)
C     WST(3) - X,Y,Z SEAT BODY AXIS SYSTEM ANGULAR VELOCITIES
C              OF THE SEAT (DEG/SEC)
```

```
C      DSA(3,3) - SEAT TO AIRPLANE DIRECTION COSINES
C      SRA      - X,Y,Z AIRPLANE BODY AXIS POSITION VECTOR OF THE
C                 SEAT REFERENCE POINT (FT)
C      RON      - SUSTAINER ROCKET FLAG (1=ON,0=OFF)
C
C   ***************************************************************
C
       DIMENSION TAE(5),F(3),T(3),XEM(3),SRP(3),UST(3),
      .          EST(3),WST(3),DSA(3,3),SRA(3)
       DIMENSION ALF(72),BET(6),AMACH(4),
      .          COEF(6),CENT(3),DES(3,3),UW(3),UWB(3),UO(3),
      .          XI(3),CONS(4),DC(3),XEMA(3),ESTIR(3),TCZ(20)
C
       COMMON /CICCAL/ ICCAL
       COMMON /COVRLY/ INST
       COMMON /CIO/ IREAD,IWRITE,IDIAG
C
       COMMON /RKTON/
      .                    ICXON(18,6,4),
      .                    ICYON(18,6,4),
      .                    ICZON(18,6,4),
      .                    ICLON(18,6,4),
      .                    ICMON(18,6,4),
      .                    ICNON(18,6,4)
C
       COMMON /RKTOFF/
      .                    ICXOFF(18,6,4),
      .                    ICYOFF(18,6,4),
      .                    ICZOFF(18,6,4),
      .                    ICLOFF(18,6,4),
      .                    ICMOFF(18,6,4),
      .                    ICNOFF(18,6,4)
C
       DATA RPD,DPR / .01745329, 57.29578 /
C
       DATA ALF /
      .      0.0  ,      5.0  ,     10.0  ,     15.0  ,     20.0  ,     25.0  ,
      .     30.0  ,     35.0  ,     40.0  ,     45.0  ,     50.0  ,     55.0  ,
      .     60.0  ,     65.0  ,     70.0  ,     75.0  ,     80.0  ,     85.0  ,
      .     90.0  ,     95.0  ,    100.0  ,    105.0  ,    110.0  ,    115.0  ,
      .    120.0  ,    125.0  ,    130.0  ,    135.0  ,    140.0  ,    145.0  ,
      .    150.0  ,    155.0  ,    160.0  ,    165.0  ,    170.0  ,    175.0  ,
      .    180.0  ,    185.0  ,    190.0  ,    195.0  ,    200.0  ,    205.0  ,
      .    210.0  ,    215.0  ,    220.0  ,    225.0  ,    230.0  ,    235.0  ,
      .    240.0  ,    245.0  ,    250.0  ,    255.0  ,    260.0  ,    265.0  ,
      .    270.0  ,    275.0  ,    280.0  ,    285.0  ,    290.0  ,    295.0  ,
      .    300.0  ,    305.0  ,    310.0  ,    315.0  ,    320.0  ,    325.0  ,
      .    330.0  ,    335.0  ,    340.0  ,    345.0  ,    350.0  ,    355.0  /
C
       DATA BET /
      .      0.0  ,      5.0  ,     10.0  ,     15.0  ,     30.0  ,     45.0  /
C
C   NOTE - BY CLASSIC DEFINITION OF TERMS, BETA HERE IS ACTUALLY PSI,
C          WHICH IS ALSO (-BETA).
C
C          THIS PECULIARITY WAS ADOPTED TO ACCOMMODATE CONVENTIONAL TABLE
C          LOOK UP ROUTINES WHICH DEMAND THAT THE INDEPENDENT VARIABLE BE
C          LISTED IN ASCENDING ORDER.
```

```
C
      DATA AMACH /     0.6  ,      0.9  ,      1.2  ,      1.5  /
C
C   ================================================================
C
C   THE AERODYNAMIC DATA HERE ARE PACKED IN OCTAL INTEGER FORM AT THE
C   RATE OF FOUR COEFFICIENTS PER COMPUTER WORD ACCORDING TO FOLLOWING
C   RATIONALE AND PROCEDURE.
C
C   1. - ALL COEFFICIENTS LIE WITHIN THE RANGE OF -1.5 TO  1.5.
C
C   2. - THE MAXIMUM POSITIVE OCTAL INTEGER AVAILABLE IN 1/4 OF A
C        COMPUTER WORD IS 37777. (16383 DECIMAL)
C
C   3. - AN INTEGER VERSION OF EACH COEFFICIENT IS CALCULATED AS FOLLOWSO
C
C        A. LET THE COEFFICIENT TOTAL RANGE OF 3.0 CORRESPOND TO THE
C           AVAILABLE INTEGER RANGE OF 16383.
C
C        B. THEN THE INTEGER REPRESENTATION IS OBTAINEDO
C
C           ICX = ((CX - (-1.5))/3.0)*16383.
C
C           THE RESULTING INTEGER IS AUTOMATICALLY STORED AS AN OCTAL
C           NUMBER AND IS AN ACCURATE REPRESENTATION OF THE COEFFICIENT
C           TO APPROXIMATELY FOUR DECIMAL PLACES.
C
C
C        C. THE VALUES OF ICML,ICMN AND ICY HAVE BEEN SET EQUAL TO ZERO
C           WHERE BETA IS EQUAL TO ZERO.
C
C
      DATA (((ICXON (I,J,K),I=1,18),J=1,1),K=1,1) /
     .06120063040670207252B,10022105451115111562B,12146130661354013732B,
     .14120143201503116363B,20234210432100620323B,20306205562112721636B,
     .23436241732464425001B,24733245452512726375B,27226273402722126737B,
     .26413257632555125657B,25656257662567625546B,25312247672434323530B,
     .23465227122217221342B,20350173141572415005B,14160134551270612137B,
     .11210102530755107003B,06304057550560405551B,05557057250604206314B
     ./
      DATA (((ICXON (I,J,K),I=1,18),J=2,2),K=1,1) /
     .06070063030656307255B,07763105431111311700B,12456127761352614247B,
     .14504146701520317120B,20245207332076520655B,20715210132122421736B,
     .23511242022461325116B,25241251412534426561B,27377274122731227060B,
     .26450260162565226076B,26174262362614226007B,25505251332446723612B,
     .23571227172221421364B,20420172621605114725B,14061132441256711737B,
     .11066102110754106723B,06221057400561305456B,05561056100622406571B
     ./
      DATA (((ICXON (I,J,K),I=1,18),J=3,3),K=1,1) /
     .06013063570706307365B,10003103751077711637B,12474131631353414212B,
     .14771155471663217613B,20360206722100521026B,21120211742173122253B,
     .23456241432471225312B,25714261562677027525B,30000277422751227235B,
     .26707262342630226514B,26665271552704026404B,25755253222467324070B,
     .23725231352223421213B,20051170441600714776B,14044132721245011625B,
     .10710101360740507164B,06431060610560105537B,05531056260615706330B
     ./
      DATA (((ICXON (I,J,K),I=1,18),J=4,4),K=1,1) /
     .06112063400662007341B,07645104611111212207B,12545132441370414414B,
```

```
.15213160141706620041B,20567211422132021507B,21612216422214722342B,
.23501242562502425675B,26357271762760730172B,30205301012761527372B,
.27136266172666627012B,27344276552756327067B,26433256552517724342B,
.24313236622243321172B,20032166221565514725B,14044131511232611576B,
.10724103070777507314B,00677062400567505532B,05513056310607206223B
./
 DATA (((ICXON (I,J,K),I=1,18),J=5,5),K=1,1) /
.07025073500761410160B,10562111611165612446B,13147137221462415441B,
.16275167502010521116B,21534221332245422572B,23032230422343224110B,
.23327241702472025434B,26237266502715227322B,27523275322753527512B,
.27331273412754630052B,30210303603057030441B,30175274002637025434B,
.25264243332350422542B,21071175021616314771B,14132134071266412240B,
.12057114241106210266B,07617073300704306741B,06641067060705707241B
./
 DATA (((ICXON (I,J,K),I=1,18),J=6,6),K=1,1) /
.11201114101177012320B,12313127341356714110B,14656154171602217073B,
.17025176352045621245B,21674223262272023345B,23706241232434224730B,
.23207234702412024564B,25066253262563426006B,26264264512667626757B,
.27067272132726027416B,27454275242776130073B,27616273002667625762B,
.26251251652406422611B,21415203321745116575B,16075155351524314437B,
.13715131361247012015B,11373111021073610635B,11051110161115611231B
./
 DATA (((ICXON (I,J,K),I=1,18),J=1,1),K=2,2) /
.04237046600530306073B,06621074651025311014B,11732127631406714705B,
.15477156771651617713B,21013211372107721007B,21111212412143021707B,
.23156236672444122521B,22721236322464425575B,26236263312626627374B,
.27031265502643126617B,27421273362721127161B,26573257472531124325B,
.25040240702313222104B,20760175461600614627B,13402125631167710732B,
.10043072040630605544B,05031044630435504424B,04457045550512205336B
./
 DATA (((ICXON (I,J,K),I=1,18),J=2,2),K=2,2) /
.04230046000517005773B,06643075271023110755B,11566127271375714715B,
.15533162041713120176B,20736210442101221126B,21257214462156522126B,
.23353240442454422603B,23130242122505425713B,26340264322767027555B,
.27116267432664127151B,27475273572725327205B,26543257572527724357B,
.25075240672320722167B,21022173311567414443B,13346123451155010720B,
.07771072220624605656B,04723044750424404275B,04253044300501405215B
./
 DATA (((ICXON (I,J,K),I=1,18),J=3,3),K=2,2) /
.04211046240524406020B,06725074371022710745B,11564125351346714546B,
.15601165671756220330B,20754207122077221153B,21354217672226722621B,
.23643244162474622740B,23572247772566726537B,27157270343027130100B,
.27572271722716427451B,27606275312745627307B,26644261432536424545B,
.25232241722330422016B,20536172441601414604B,13423124011145010545B,
.07676071050633605654B,05136044340433704272B,04311045550501305163B
./
 DATA (((ICXON (I,J,K),I=1,18),J=4,4),K=2,2) /
.04333046770530105751B,06546073451010411037B,11673125601347214526B,
.15524165751747520343B,21021211522101721436B,21741222572255423067B,
.23754240522546723341B,24326255002655327326B,27656276153055030406B,
.30024275702766230027B,27724300002774727546B,27115264202565424753B,
.25512245152341322135B,20457171571604714701B,13540125231146310515B,
.07643072700647606005B,05320047250441204431B,04510046510505105255B
./
 DATA (((ICXON (I,J,K),I=1,18),J=5,5),K=2,2) /
.05301056050624206541B,07235077711073211574B,12516132471421715067B,
.15774167471752520411B,21206217302231622705B,22713230452364524445B,
.23530245422544526371B,27077274563005330270B,30476305473046030373B,
```

```
.30261302363024530212B,30205301753010730017B,27734273772670225661B,
.26165250072365222534B,21230175161610714721B,13717131211234311437B,
.11030104571002507234B,0005062010577005666B,05600056600600306172B
./
 DATA (((ICXON (I,J,K),I=1,18),J=6,6),K=2,2) /
.07531076121012410273B,10570112261176512466B,13261142021475015442B,
.16255171115200762073 4B,21354216642223122624B,23343240042442525154B,
.23700243132467625343B,25556261452646326671B,27245274062757027777B,
.27724276602765427605B,27533275322737427147B,26772265102576525101B,
.26247252202403322611B,21476202541723116355B,15602146301415413613B,
.15100123631170411140B,10645103531011710025B,10042100641002310035B
./
 DATA (((ICXON (I,J,K),I=1,18),J=1,1),K=3,3) /
.02543030060342104127B,05075060020700210066B,11121120371301614022B,
.14075153311606217047B,20326211502157322140B,21050217362205522435B,
.24772254732620726007B,26316272703026431010B,31222310243126631217B,
.31027306353065730511B,30353302753013727631B,27145263672546324460B,
.25241242602323622122B,21024175671640415112B,13611123741107507773B,
.06775061130532504366B,03413030200306603415B,03505040210444205033B
./
 DATA (((ICXON (I,J,K),I=1,18),J=2,2),K=3,3) /
.02545031450347404245B,05070060260677410070B,11007117641272513711B,
.14521152741620117274B,20306212652163122103B,22262222122245123051B,
.25050256542633525707B,26422274723047531216B,31426310703155531412B,
.31256307443052430405B,30312302073006327526B,27060263612546724521B,
.25217242502325322141B,21004175621623614564B,13245120671100007705B,
.06726057230504104411B,03550320503075031548,03410040160444405056B
./
 DATA (((ICXON (I,J,K),I=1,18),J=3,3),K=3,3) /
.02734033210367004321B,05222061130702110007B,11010120241271413650B,
.14542153631650017513B,20336215132170622070B,22114222632260423254B,
.25161261422677325664B,26737300303074531532B,31706314613230332044B,
.31465311503070330532B,30434302653012427540B,27116263522554424620B,
.25327243172321722150B,20741174461611014556B,13265120661065707670B,
.06700057730513704332B,03732034350330003315B,03553041440444405075B
./
 DATA (((ICXON (I,J,K),I=1,18),J=4,4),K=3,3) /
.03162034010401204451B,05246061740707510131B,11056120001267413636B,
.14570155171652217557B,20525211712172222151B,22440227572332223603B,
.25254263652737626413B,27365303543126731773B,32136320553240532042B,
.31545313423107030625B,30504303673006227521B,27111264362565725105B,
.25465244232326222163B,20713173071602114512B,13313121461073507636B,
.06637060000533404615B,04156036300347103456B,03716042760461405171B
./
 DATA (((ICXON (I,J,K),I=1,18),J=5,5),K=3,3) /
.04420045520477205375B,06102067140770310666B,11602125001344014322B,
.15160161021712520122B,20705214642222122762B,23224233122376224526B,
.25223261772713327725B,30535312163161131764B,31723317323177131664B,
.31474313303112730700B,30407302503010727660B,27427270312632125501B,
.26022246172340022217B,20742173021600114466B,13372123371133610476B,
.07710072250661006173B,05725054570525605302B,0536605+320561506074B
./
 DATA (((ICXON (I,J,K),I=1,18),J=6,6),K=3,3) /
.06674067650735307361B,07772105311123011733B,12654135201425015132B,
.15731166141747620413B,21063214632204022462B,23167236472432425055B,
.24373251432566726401B,27040274212765230110B,30225303543044630512B,
.30522304663031530115B,27660273732713626755B,26447260622540324552B,
.25554245022332722146B,21027177611674615775B,15063142541352513032B,
```

```
.12262116331116210554B,10225077750764607547B,07512075540770307757B
./
 DATA (((ICXON (I,J,K),I=1,18),J=1,1),K=4,4) /
.02231024720305703712B,04731060510713510207B,11320122321315614035B,
.14655154071611016773B,20163212032200722426B,22717231542363323703B,
.25313261102652026743B,27145277753063031350B,31664317433161332254B,
.32245320743162131306B,31012306373033427744B,27275264572554124613B,
.25051237712276221722B,20710176071653715415B,14260127771156310274B,
.07036060130510404202B,03250025360231302351B,02466030760346604112B
./
 DATA (((ICXON (I,J,K),I=1,18),J=2,2),K=4,4) /
.02252025130514603707B,05017060770704110230B,11163121541312113743B,
.14615154241622617222B,20216211622202622466B,22727232612402224222B,
.25362261342662326000B,27216301073070031517B,31756317573163432320B,
.32245320713153331214B,30765305243025227674B,27251264272555424711B,
.25057240332302721752B,20652176141646015174B,13754125561137710102B,
.06757060220513304131B,03274026030240602422B,02550031170350604053B
./
 DATA (((ICXON (I,J,K),I=1,18),J=3,3),K=4,4) /
.02413027300334404046B,05076060720712110100B,11112120701300113660B,
.14570155071645117470B,20370211552211722504B,23075235612407324431B,
.25472263522711426404B,27356302673113432011B,32203322233254732461B,
.32341320673153731266B,31070306513034427764B,27312265212567125013B,
.25173241212303721736B,20627174651631415107B,13733125061117610030B,
.06740060050514204215B,03411027530264102643B,02733032020356204236B
./
 DATA (((ICXON (I,J,K),I=1,18),J=4,4),K=4,4) /
.02612030450342104107B,05033060240707410121B,11100117641271013560B,
.14663157271653017526B,20505212752214122655B,23346237542433324574B,
.25522265422745727142B,27616305543146032140B,32442324053265232524B,
.32403321753166031400B,31217310463046130004B,27356266242607625234B,
.25363241672304221764B,20612174051615615027B,13702125021127410060B,
.07022061020525504365B,03635032250313303101B,03137033630377204370B
./
 DATA (((ICXON (I,J,K),I=1,18),J=5,5),K=4,4) /
.04377046010517305704B,06470074101035511365B,12340132151411314710B,
.15603165671755520473B,21331221302264023320B,23625240442423224412B,
.23234234662376224440B,25204257342640427014B,27205277223022230543B,
.31737315403131531141B,31037305773046630205B,30210275302677126144B,
.26656255312445523330B,22020205741721515614B,14426132221220511175B,
.10412074550660060031B,05243050710467704521B,04415044750465205103B
./
 DATA (((ICXON (I,J,K),I=1,18),J=6,6),K=4,4) /
.06557066520707507421B,10160106671146612277B,13152140131470515605B,
.16371172762006520621B,21377220772254023135B,23432236762416524454B,
.24254247062522725603B,26207270272745730110B,30246307703111531037B,
.30753306543053430375B,30074300462767427370B,27122266012615125356B,
.26173251432425323153B,22112210351773216650B,15715147041401213300B,
.12440116151110110355B,07704073160713007041B,07003070320720307400B
./
 DATA (((ICYON (I,J,K),I=1,18),J=1,1),K=1,1) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
```

```
 DATA (((ICYON (I,J,K),I=1,18),J=2,2),K=1,1) /
.21153205252061220746B,20756210072151222170B,22323217532220522207B,
.22101216502157121042B,20147174421654615744B,16200165551703017244B,
.17020174021773520436B,21143213372122421254B,21155210032076020774B,
.21012205662030620424B,20474205562115221543B,21634216122140621327B,
.21637214302122321267B,21473215472126220760B,20472204712020520066B,
.17506174452000320221B,20105202502030320025B,17753172721776420402B
./
 DATA (((ICYON (I,J,K),I=1,18),J=3,3),K=1,1) /
.21453217472204022310B,22470227312325323562B,24142240462434724361B,
.24370237012274021767B,21046200621727416652B,16710174401716017314B,
.17731203162102021572B,22461230772316122670B,22402220742167221523B,
.21243210442074721157B,21221214342213323224B,23422233323144227706B,
.23032226702311123220B,23334232332271322000B,21431212142102420623B,
.20362203532055221071B,21132211602104420743B,20652206752105221112B
./
 DATA (((ICYON (I,J,K),I=1,18),J=4,4),K=1,1) /
.22743231462342423725B,24132243102447625021B,25546257012601725752B,
.25537251242441025250B,22422212462027717673B,17656200502014220252B,
.21047215662230623113B,23617242502450124001B,23456231732262722344B,
.22061216532137521612B,22005223562303724016B,24367243652423624071B,
.24064241362461224727B,24671245412355123036B,22542222572206622010B,
.21607215332166522035B,21736220542211222075B,22133221222222322343B
./
 DATA (((ICYON (I,J,K),I=1,18),J=5,5),K=1,1) /
.27216272562741430077B,30603307743124131526B,31625317113177031553B,
.31153304212773327043B,26053252512461224450B,24376243412432524216B,
.25630250732523625355B,25510256252551025463B,25405252052514225114B,
.24755245412444624410B,24546250672554025742B,26603274353031230630B,
.31450313253127130476B,27555276412740727113B,26720266362657526550B,
.26640267042672226611B,26034265772654526541B,26407265122066027012B
./
 DATA (((ICYON (I,J,K),I=1,18),J=6,6),K=1,1) /
.34066342343447434435B,34313341533414533452B,33535337553375233601B,
.33430533003307032524B,32210315753127731153B,31016306443043730227B,
.30024305063070030625B,30475302163015130145B,30143300002776430012B,
.27734277312765327644B,27740301503052231123B,31554321723261532744B,
.33601335373331033402B,33401332413307033045B,33041331033325033330B,
.33356333133332733250B,33306334023342333503B,33643337113371233735B
./
 DATA (((ICYON (I,J,K),I=1,18),J=1,1),K=2,2) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICYON (I,J,K),I=1,18),J=2,2),K=2,2) /
.20656206562070521030B,21213213552145721720B,22030217602162321353B,
.21336213422077520336B,17413167371653016551B,16773171161715617405B,
.17257176362025520374B,20445203742050020705B,20631207562120321030B,
.20522204512042120457B,20416207202112321153B,21313213142120121347B,
.21634215042145721420B,21372212432071420537B,20361203062037420127B,
.17745200121752520045B,20264203272021120124B,17711177742015720212B
./
 DATA (((ICYON (I,J,K),I=1,18),J=3,3),K=2,2) /
.22030222222245422472B,22636230502322023341B,23356233032325023076B,
```

```
.23067224412154720750B,20337175121725617072B,17356171531713317376B,
.20347207562152621640B,21631220002227222617B,22463225772252122106B,
.21650213352117021447B,21566217562221122444B,22462224462251122644B,
.23355233112317723045B,23003223752170121463B,21217210012065720452B,
.20335205062075421073B,21144212172112620756B,20774211032124721365B
./
 DATA (((ICYON (I,J,K),I=1,18),J=4,4),K=2,2) /
.23506236432407024350B,24406244322451624730B,24775247222460624465B,
.24334237462312622140B,21402205442006117767B,20123202402027320451B,
.21705223412275223446B,23566237722434524773B,24747244252365123367B,
.23015224642212622345B,22503227622331123507B,23635236572361424036B,
.25017250172465324372B,24076235752343623100B,22523221652174321601B,
.21504215012220222173B,22274223122226322255B,22357224302255722713B
./
 DATA (((ICYON (I,J,K),I=1,18),J=5,5),K=2,2) /
.30241303273030130347B,31037311503107731361B,31401313653123430710B,
.30345277562707426340B,25522251022467124551B,24727251132530425400B,
.26237265122675427117B,27205273062726427325B,27227271012663626331B,
.26053255152546125522B,25713260272630526527B,26701267602736230342B,
.31721317263112030370B,30346303413000427465B,27313272342711127122B,
.27111271532720327120B,27106270712702326737B,26066267532723027420B
./
 DATA (((ICYON (I,J,K),I=1,18),J=6,6),K=2,2) /
.34727347253463434505B,34623344273441334145B,34234343763442734260B,
.34023335563323732716B,32460321003174131713B,31563314763135231257B,
.31620316373165431650B,31632316163152131526B,31733316353140231410B,
.31360314343152131554B,31521316333167731655B,32026323443271633206B,
.34326345233450634454B,34421341013371333637B,33641337063371434117B,
.34211342413430234321B,34222342413416634027B,34046337623406334147B
./
 DATA (((ICYON (I,J,K),I=1,18),J=1,1),K=3,3) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICYON (I,J,K),I=1,18),J=2,2),K=3,3) /
.21341214302146521477B,21547216132161321702B,21761217452200322070B,
.22067216772157221436B,21334207262037720341B,20272175571735517455B,
.21137212462150521422B,21264213072157221733B,21707220032155721521B,
.21352212012111421126B,21133212372134621277B,21277212662124121214B,
.21302211542115421127B,21147212002117421153B,21040206632071320663B,
.20537202522017420502B,20534205062045620356B,20225202522033720321B
./
 DATA (((ICYON (I,J,K),I=1,18),J=3,3),K=3,3) /
.22646227512305023106B,23272234472342423320B,23452232702327423325B,
.23337232602305022613B,22342215772105020672B,20734203612011720061B,
.22254225552305623164B,23235235212412424340B,24340241032345023120B,
.22665225712226522360B,22414226312271322720B,22741226402247422304B,
.23102230232270122603B,22626226352271322616B,22343221162174121654B,
.21570214402146121572B,21630216432161021544B,21412214262146121440B
./
 DATA (((ICYON (I,J,K),I=1,18),J=4,4),K=3,3) /
.24403245012466424766B,25065251102516525210B,25235251242475624771B,
.24752247232447024017B,23462224572207421442B,21346211522114621313B,
.23516240562450025340B,25460257032620326363B,26352260122516724621B,
```

```
.24441242602373423700B,24620242062421324162B,24136237362364623465B,
.24714246702453224350B,24350243312414224173B,23762235002332323222B,
.23164232372314123042B,23067230362306722755B,22711226512267022732B
./
 DATA (((ICYON (I,J,K),I=1,18),J=5,5),K=3,3) /
.31206312243126131465B,32030317163156431426B,31436314043127731071B,
.30642303242770427245B,26564262342602025632B,25475254652544625553B,
.27456277053005130161B,30524310363110331166B,30765307073027630122B,
.27570273532731427136B,26777270612717527204B,27223271442727530046B,
.31426314533104030630B,31000307653063630517B,30421305003060430620B,
.30566304573034230135B,27754276602747327246B,27163271242714127263B
./
 DATA (((ICYON (I,J,K),I=1,18),J=6,6),K=3,3) /
.35511352543514434727B,34707346213451434474B,34416343213440434106B,
.33663333533302032464B,32277317733147531327B,31260312213116331175B,
.33123330743307333033B,32774327023256032666B,32630325273242132312B,
.32213321643215632157B,32162320603210132071B,32016321233251332725B,
.34076342023417334037B,33620336043360633575B,33620336613402134170B,
.34303343033417034103B,33743336223346233257B,33146331663315733216B
./
 DATA (((ICYON (I,J,K),I=1,18),J=1,1),K=4,4) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICYON (I,J,K),I=1,18),J=2,2),K=4,4) /
.21467215032144721511B,21504214232147421473B,21475215502164321635B,
.21530214572146521325B,21202207752057520455B,20606205372054020501B,
.21310214502156421527B,21512215502166221746B,21751220142177321626B,
.21577215712153521520B,21515214652150321530B,21555214772135021257B,
.21251211602112621055B,21003211172123021236B,21254212702127721173B,
.20774207722104321047B,21021210442103621043B,21057210432065520670B
./
 DATA (((ICYON (I,J,K),I=1,18),J=3,3),K=4,4) /
.23175231662322123247B,23342233362323323300B,23373233602336623324B,
.23170231152305122706B,22517223152174421576B,21726216622152021175B,
.22556227312320423203B,23206233022346523670B,23704236462350523232B,
.23130231142304023045B,23043230762310623065B,23057230122265522431B,
.22761226612252622414B,22471225712272123001B,23004226762260522474B,
.22304222502230622400B,22405224242237322344B,22321222762223222165B
./
 DATA (((ICYON (I,J,K),I=1,18),J=4,4),K=4,4) /
.24644247252477725013B,24737250532507525044B,25046250202471324640B,
.24470244132447724202B,24002234702306422724B,23057230442261022320B,
.24005242312450425055B,25170252332541025556B,25673255322522025017B,
.24712246712465524705B,25016247472462424527B,24447243622424724063B,
.24503244002420724033B,24066242002424424226B,24232242712415523734B,
.23615235722364223717B,23744237172370623626B,23572235612350123414B
./
 DATA (((ICYON (I,J,K),I=1,18),J=5,5),K=4,4) /
.31407315063157031740B,32012317243161131564B,31660316633157631474B,
.31344310713052130124B,27500271522674626526B,26220257702574726037B,
.27511275132755127737B,30126302153025530302B,30372304413044630450B,
.30356300332746527417B,27435275112770330230B,30472306303071030643B,
.32052317023142631444B,31324312143112531063B,31043312053114231367B,
```

```
.31365314263144431431B,31345310703075330675B,30647305053042330405B
./
 DATA (((ICYON (I,J,K),I=1,18),J=6,6),K=4,4) /
.35644356133552735377B,35372353043522335340B,35313351723505334751B,
.34540342763402533607B,33245332473236132072B,31573313253120431174B,
.32717330243314733240B,33372335113356633661B,33714336573345133216B,
.33071327313264132620B,32615327043303433072B,33151332573341733577B,
.35062351233520235217B,35113350713504035007B,35032350333516535243B,
.35441355623557735554B,35520354673534135213B,35057347103472334556B
./
 DATA (((ICZON (I,J,K),I=1,18),J=1,1),K=1,1) /
.23376224452166320756B,17742166421562614734B,14151134201272212257B,
.12024117731142311250B,10712113011150211704B,11732117741177612043B,
.11363113701127011446B,12230136061503415542B,16116164551676117410B,
.17730202372004417647B,17601176342023320611B,21337223352325624172B,
.22713233712346223627B,23700242062447724702B,25007246732462425037B,
.25161251672525024750B,24540245462446524222B,24001235362326422736B
./
 DATA (((ICZON (I,J,K),I=1,18),J=2,2),K=1,1) /
.23406224622172321021B,17734166331571514714B,13750133041251312170B,
.11667116431155210643B,10753111341124111401B,11544116301166211662B,
.11141112251132611412B,12037132241464515413B,15745163651674617366B,
.20047203712003117447B,17416176172011220677B,21442234223234241338B,
.22656234212356023763B,24026242122435125031B,25151251302501625124B,
.25172252532521225113B,25040246562450724266B,24003235762327022746B
./
 DATA (((ICZON (I,J,K),I=1,18),J=3,3),K=1,1) /
.23327225412167420770B,17722166751566314621B,13656131231243212013B,
.11673116441100510412B,10560107631114111162B,11257114661141111352B,
.11120112061126011353B,11717126311367214624B,15452161771670417350B,
.17723201301736217267B,17237172341770320721B,21505223312315423741B,
.22734233232373324255B,24414243642447125101B,25364253332532425316B,
.25423253502523125067B,25026247332460224303B,24042236062340323014B
./
 DATA (((ICZON (I,J,K),I=1,18),J=4,4),K=1,1) /
.23344226752176620767B,17767167611575015037B,13753132171237612050B,
.11623115421063010311B,10216105711066710667B,11055113001130511444B,
.11053111171120711357B,11625122771327314304B,15250161341672617343B,
.17670201231746017360B,17240173171773120606B,21305220702276623634B,
.22740233602370124250B,24461246442505625335B,25451255132556325530B,
.25520254572536225171B,25030247132461724416B,24167237412350023063B
./
 DATA (((ICZON (I,J,K),I=1,18),J=5,5),K=1,1) /
.23350224542172421060B,20150171301612415177B,14356136601306112354B,
.11675113751072510513B,10553105201072011106B,11236112521150211763B,
.11444115631177012240B,12557132731405714756B,15714166641751220143B,
.20414204002035220131B,20023177621775520240B,20555212352201122561B,
.22127225702324423705B,24607251712544125642B,25651256142557425544B,
.25411254072533625315B,25233251112474424443B,24210237542347423043B
./
 DATA (((ICZON (I,J,K),I=1,18),J=6,6),K=1,1) /
.22740225632215321303B,20566177021670016002B,15314146671425113460B,
.13306127601265012400B,12227122021224712252B,12256123421244112542B,
.12644130411330613542B,14152147161544116051B,16432170571746217773B,
.20315205152051220543B,20637206432021620077B,20272206002117221660B,
.21234221002261323250B,23646241252416624311B,24424245312454524534B,
.24547245252442724357B,24316241712400023600B,23114227552254622365B
./
```

```
 DATA (((ICZON (I,J,K),I=1,18),J=1,1),K=2,2) /
.24077233362247321525B,20437172631611315005B,13774126341162111326B,
.11075106051030510136B,10221105731101511153B,11251113011133711405B,
.10424104431036611130B,12264143471632717241B,17570201502052220334B,
.20554206342074720315B,17373176762025220661B,21350221302566233364B,
.23031235552376123550B,23740242632477725347B,25573255432556725676B,
.26110262442642426355B,26041256252534125006B,24440241222353423226B
./
 DATA (((ICZON (I,J,K),I=1,18),J=2,2),K=2,2) /
.24053232452243521522B,20357172271610114776B,13676126571170111254B,
.10754106451025507715B,10160105441072511025B,11164111311115711270B,
.10340104021047011156B,12322142161622617127B,17464200411761720137B,
.20353204622055220027B,17414177452033120714B,21377221372266323330B,
.23023235252406524056B,24237244522515525604B,25752257642570325743B,
.26017261122634626256B,26067255462534525041B,24566242352361623272B
./
 DATA (((ICZON (I,J,K),I=1,18),J=3,3),K=2,2) /
.23746232262237021450B,20323171771600314631B,13522125641171111263B,
.10770105701021607776B,10034103331050010557B,10723107251076211030B,
.10307103361045211244B,12443140001557516522B,17167176531737717774B,
.20330203762034017614B,17511177672032120651B,21273217672256223326B,
.22751234372404524406B,24634250522531625574B,26056261022613026145B,
.26262262612626126206B,26061257332542125126B,24611242772304623247B
./
 DATA (((ICZON (I,J,K),I=1,18),J=4,4),K=2,2) /
.23710231352231221426B,20244170651574614770B,13643127021174711265B,
.11117106241015010067B,07777100601030110374B,10576106601073611044B,
.10317104111050411260B,12263136171512615754B,16573174041715617606B,
.20144202051745517365B,17632200152025020650B,21352220462253223254B,
.22747233462401024573B,25157254222557025733B,26061262402632726266B,
.26325263272627726163B,26017256302545625140B,24614242662367323216B
./
 DATA (((ICZON (I,J,K),I=1,18),J=5,5),K=2,2) /
.23537227132204121434B,20554176021640215425B,14401134421262512145B,
.11465110571061610431B,10365105051047710660B,11071111171121511371B,
.11042111671137111631B,12203127001345014272B,15212160751704317614B,
.20354203452014720265B,20375204512057620745B,21077214102200322564B,
.22306230362373124405B,25062254572573426132B,26173262312621126276B,
.26236261732613025754B,25530253142506524701B,24455241242351023042B
./
 DATA (((ICZON (I,J,K),I=1,18),J=6,6),K=2,2) /
.23346227112231121677B,21066202041740116451B,15655150121421713533B,
.13107125521230512110B,12021121011212412045B,12001120051215312326B,
.11604120261232212661B,13315137531442115120B,15626163261665417264B,
.17730203162065621012B,21174213032150721626B,21414213612170522307B,
.21660223132300723457B,24062244702463725006B,25115253132531525263B,
.25215251462505224755B,24524243202407223676B,23404232162275522411B
./
 DATA (((ICZON (I,J,K),I=1,18),J=1,1),K=3,3) /
.24102233152242221431B,20212170241564614471B,13410123241132710565B,
.10341101220763207454B,07300075520761607746B,10330104661064011040B,
.07676076030763510003B,10602122531411115224B,16014164521707317457B,
.17721200321773317442B,17663202132055721170B,21627222312256223102B,
.23333237202404724017B,24162246722525325723B,26326265772674327173B,
.27251273002734727253B,27101266412620625506B,25140244372376623323B
./
 DATA (((ICZON (I,J,K),I=1,18),J=2,2),K=3,3) /
.24064232542240521421B,20307171131563614524B,13403123521141010677B,
```

```
.10366102371000307454B,07243073400764010061B,10203104711053710656B,
.07625075070763410017B,10634122621370415065B,15665164361067717305B,
.17614200571757117571B,17752202452057021167B,21523220742250422761B,
.23253230652407224156B,24462250162553126322B,26670270152710127224B,
.27330273472736627223B,26766264362614425607B,25250245462404723356B
./
 DATA (((ICZON (I,J,K),I=1,18),J=3,3),K=3,3) /
.24055232472231421335B,20243171151567414513B,13400123271141710756B,
.10534104671017307424B,07270070770746110064B,10307104561054210635B,
.07557075500761310074B,10654120601336614427B,15513162471623117112B,
.17556203751762617745B,20043203252061621156B,21451220012234722664B,
.23155235752413524465B,25060254402605726422B,26746271252724527255B,
.27260272742726727234B,26742264502610325553B,25205245372402223304B
./
 DATA (((ICZON (I,J,K),I=1,18),J=4,4),K=3,3) /
.23711231302221521243B,20102170001557514605B,13523124421163511164B,
.10770106561014607547B,07273074230762310003B,10201103031044610631B,
.07565076140776310202B,10701117221276113722B,15051156361572616636B,
.17412200362004020113B,20300204732077021223B,21445217162224322636B,
.23055234742424424762B,25451260542636626611B,26743271002727427355B,
.27403273242716627124B,26710264052610525570B,25151244502376023207B
./
 DATA (((ICZON (I,J,K),I=1,18),J=5,5),K=3,3) /
.23442226462200221210B,20343174301635015303B,14255133561253112037B,
.11433110121045110172B,10152101371024610345B,10533106661101211152B,
.10243103341051611047B,11326117341234713170B,14204151011576716564B,
.17346200002037320655B,21056212212132621404B,21542216722210322461B,
.22573233262421324056B,25321257672625626600B,26720270472716227231B,
.27142270112657026313B,26063256322534725025B,24505241402350423006B
./
 DATA (((ICZON (I,J,K),I=1,18),J=6,6),K=3,3) /
.23164225422220421404B,20751201631732416477B,15641150161416413515B,
.13151125611231012047B,117271117061172011712B,11664117301204512252B,
.11341114751164612141B,12466130551354314171B,15004155031617416646B,
.17274177232035320730B,21171214232160021702B,21776221052230222574B,
.22104226572334523766B,24353246702510625336B,25534256542566625627B,
.25564254402533325133B,24664244342422424010B,23572232462274322401B
./
 DATA (((ICZON (I,J,K),I=1,18),J=1,1),K=4,4) /
.23675231522227121260B,20063166601562014572B,13474125141175111504B,
.11161105671015407655B,07355073200730707302B,07463075520764610202B,
.07464074610750310036B,10707120431341414441B,15226157221642417022B,
.17252175301776320224B,20373206062114421501B,22125226062306223242B,
.23265230012375524003B,24170245472511125502B,26103265432707427256B,
.27440275762753727457B,27336270312644626031B,25431250002423523534B
./
 DATA (((ICZON (I,J,K),I=1,18),J=2,2),K=4,4) /
.23723231612224021317B,20161167611562714530B,13470125331202111555B,
.11260107651037407776B,07425072720726207226B,07364074610760110061B,
.07421074540747010064B,10727120441330314321B,15171157261635316666B,
.17200175212003320166B,20421206242115221474B,22075223732261623043B,
.23212236002402624151B,24407247312536226127B,26570270772730427475B,
.27561276342760327525B,27341267772643426046B,25463247742431123610B
./
 DATA (((ICZON (I,J,K),I=1,18),J=3,3),K=4,4) /
.23741230602215621216B,20100167731562314512B,13505125161205111650B,
.11521112121067310145B,07434071630707107140B,07271074000765210122B,
.07442074660755310105B,10661116641302713761B,14750155221607116474B,
```

```
.17125175502014120307B,20516207732125721520B,21756222652253422736B,
.23141235312405124350B,24734253532575226333B,26665272052743627501B,
.27574276232760127473B,27262267512635426002B,25437250032426123562B
./
 DATA (((ICZON (I,J,K),I=1,18),J=4,4),K=4,4) /
.23670230342211221070B,17705166661566414637B,13617127041222512006B,
.11771113631071210227B,07520072100712007205B,07266074300763110147B,
.07507075470761610110B,10650116171246413332B,14336152551562116341B,
.17047175422023320566B,20736211222133121551B,21745222722226523027B,
.23055234652417724575B,25212255602624226570B,27075273172746627550B,
.27613276352755227427B,27177266472630425734B,25371247232416423540B
./
 DATA (((ICZON (I,J,K),I=1,18),J=5,5),K=4,4) /
.23026222622214412054 3B,17605166211561614522B,13527126621202011331B,
.10575100620756707366B,07274073110735707462B,07635100021010010231B,
.10513113231203412563B,13404142071476715532B,16303166771736717660B,
.17775203412063721036B,21443214732167722156B,22255222571230372332 5B,
.22740233672403624516B,25166256772633726740B,27171273342737427437B,
.27320272742714626670B,26452260542553425215B,24721243542373223250B
./
 DATA (((ICZON (I,J,K),I=1,18),J=6,6),K=4,4) /
.22531220612135620605B,20030172431640415510B,14604137601317612514B,
.12056115121123410770B,10660106251064410704B,11045112421141011532B,
.11077114071175212537B,13021135101406614465B,15176155421624316747B,
.17451200712050421030B,21363215532206722245B,22425225372261723021B,
.22375230342337523752B,24335247122523625475B,25657260232606426027B,
.26005257412560125477B,25255247752454724254B,23731233632276522353B
./
 DATA (((ICLON(I,J,K),I=1,18),J=1,1),K=1,1) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICLON(I,J,K),I=1,18),J=2,2),K=1,1) /
.20042177501775617756B,17775200222012220165B,20174201242015720126B,
.20156201522035120552B,20307202172020620153B,20041200352006720116B,
.20230202022015720121B,20060200442002017745B,17660176261762017640B,
.17703176611765317576B,17635177342021620442B,20574205442047320522B,
.20526203072006620063B,20320203742030420314B,20303203262026320162B,
.20102200342002420062B,20125201752013620013B,17733176131765517761B
./
 DATA (((ICLON(I,J,K),I=1,18),J=3,3),K=1,1) /
.17744200332004020046B,20073201322021220206B,20224202372034220450B,
.20420205012074020724B,20542204032031520252B,20223201722020420200B,
.20352203372031620276B,20212201602004217714B,17577175161750217471B,
.17454174771706417757B,20067202552054321107B,21203212422121021177B,
.21023205512046420454B,20501205702062520606B,20607206022053720472B,
.20343202262015120300B,20317203402024620204B,20121200201775617731B
./
 DATA (((ICLON(I,J,K),I=1,18),J=4,4),K=1,1) /
.20076201352015220176B,20166201312006120067B,20227203742046720532B,
.20664207052104020755B,20610204012032320243B,20225202212020520140B,
.20367204442044520356B,20271202342007117636B,17534175221744517451B,
.17451175421775720140B,20347205752077721417B,21477215032145521361B,
.21150210562107121074B,21126211372111021065B,21111210072100321017B,
```

```
.20675205232044520432B,20377204222041520350B,20321202462015320102B
./
 DATA (((ICLON(I,J,K),I=1,18),J=5,5),K=1,1) /
.20375203332036620410B,20451205432036620443B,20417205202061021004B,
.21061210402111521073B,20706205222034320232B,20206202402020020115B,
.20337203252032220277B,20263202202016220101B,20103200772013720174B,
.20175203572042120572B,20760212222150521645B,22147224042255322561B,
.227[illegible]4226472260422435B,22173222142222122235B,22237222272217622147B,
.22006220332177221632B,21511213712125721143B,21033207142063420571B
./
 DATA (((ICLON(I,J,K),I=_,18),J=6,6),K=1,1) /
.21566214472142121463B,21222210672076120733B,20673207052076321040B,
.21143211602112221067B,21010207052060220523B,20407203172025220167B,
.20424205162054320537B,20506204462044520503B,20523205422060020657B,
.20716210452122221355B,21521217012212422423B,22623230172322423274B,
.23652235732347123444B,23533232532325223212B,23167231702316123216B,
.23100231042304322735B,22637225462244522320B,22155221112201521756B
./
 DATA (((ICLON(I,J,K),I=1,18),J=1,1),K=2,2) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICLON(I,J,K),I=1,18),J=2,2),K=2,2) /
.20004177532000017766B,20004200762012720155B,20146201052017220107B,
.20177201612025620253B,20171200111773417740B,17753200032002520053B,
.20203202152017417751B,20007200112002520033B,17737177431762117610B,
.17525176151762417655B,20007200722022220467B,20601206072054620553B,
.20477203602016020105B,20102201042016320222B,20234202632026320154B,
.20067200552003020074B,20111201012005520003B,17726177341772517736B
./
 DATA (((ICLON(I,J,K),I=1,18),J=3,3),K=2,2) /
.20031200622013120103B,20053201362011520127B,20145202122032320434B,
.20514204612041020271B,20167200532000420021B,20025200532007720066B,
.20346204022043420116B,20220202002015620124B,20020177641751117447B,
.17555177222003320166B,20334204422065021133B,21230212062115721210B,
.21132207162046620335B,20325203452043120504B,20525205062045020355B,
.20267202422017620213B,20235202222016520125B,20115200602004220041B
./
 DATA (((ICLON(I,J,K),I=1,18),J=4,4),K=2,2) /
.20214202222024720307B,20256202002021620164B,20166203132045320551B,
.20613206042051420307B,20222201122003220031B,20067201242013220100B,
.20374204562050320366B,20516204452037520257B,20116200261762217562B,
.17710200512031520464B,20575210002122221357B,21473215112145321416B,
.21535212762107120733B,20665206742075420763B,21001207722071320663B,
.20567204542036620333B,20352203562033620331B,20326202742024720244B
./
 DATA (((ICLON(I,J,K),I=1,18),J=5,5),K=2,2) /
.20723206702063220600B,20561205522054420530B,20536206772070420745B,
.21036210232100020650B,20507203012015020073B,20126202072021320216B,
.20444205402057120551B,20535205202045620446B,20437204462046120455B,
.20452205472101321113B,21267214552164022000B,22121221712232122430B,
.23054227302241022226B,22214222372222622224B,22241222302215622110B,
.22025217702171221577B,21515214252132021166B,21051210062104521045B
./
```

```
 DATA (((ICLON(I,J,K),I=1,18),J=6,6),K=2,2) /
.22020217332165121566B,21507214472133521310B,21243212262124221325B,
.21356213432132521202B,21073207542066420633B,20600205442051020440B,
.20717210242105121050B,21031210412105121100B,21125211702123121273B,
.21362214702162121762B,22076222272235422444B,22524227052305323146B,
.23540235442350123443B,23355232032316023211B,23227232062317423172B,
.23157231352307423024B,22747226752255522416B,22263221642214722142B
./
 DATA (((ICLON(I,J,K),I=1,18),J=1,1),K=3,3) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICLON(I,J,K),I=1,18),J=2,2),K=3,3) /
.20036200222003220045B,20065201272015320176B,20177202422026420270B,
.20260202022017120201B,20130200672007020063B,20057200612007320111B,
.20247203122027520126B,20146201332012720071B,20020200241772017751B,
.17751177622013720214B,20274203512047120600B,20605205672051620451B,
.20361202152012220061B,20002200122007520170B,20222202112020120163B,
.20144201072003020034B,20020200232003120020B,17746177631775217761B
./
 DATA (((ICLON(I,J,K),I=1,18),J=3,3),K=3,3) /
.20142201722017520145B,20225203012030320274B,20325203642041320427B,
.20437203612032720301B,20200201752007420071B,20115201202011220120B,
.20452205312053520364B,20420204432037320273B,20171201071774217745B,
.20027201352035620504B,20634207442111021213B,21273212602115521017B,
.20761205302034720244B,20210202432033520424B,20474205022046220460B,
.20423203342024120211B,20155201562012720113B,20062200612007120062B
./
 DATA (((ICLON(I,J,K),I=1,18),J=4,4),K=3,3) /
.20322203352036420354B,20374204152041420352B,20421204752053620564B,
.20574205452055520425B,20333202072012020054B,20070201072012320173B,
.20546206152063720703B,20661206672055220451B,20333202522013720160B,
.20261203652053720752B,21135213462146121543B,21611215412145421350B,
.21313211012071220575B,20555205752064520720B,20767210122077320741B,
.20672206132046220366B,20337203122027620255B,20244202352024020235B
./
 DATA (((ICLON(I,J,K),I=1,18),J=5,5),K=3,3) /
.21153211342115021160B,21154211162104320776B,21002210602105121045B,
.21047210152073220627B,20440203172023120144B,20112201402016320226B,
.20622207132071620704B,20745207702074220712B,20705207122066620722B,
.21015211272124521345B,21470216412201622117B,22155222322227422367B,
.22561224572222222136B,22154221712221522237B,22236222332220022153B,
.22121220372174321635B,21521214112132021213B,21131211062107521107B
./
 DATA (((ICLON(I,J,K),I=1,18),J=6,6),K=3,3) /
.22232221122204722020B,21673216172155021446B,21412214002145121437B,
.21416213422126321172B,21055207122062320577B,20564205522054320534B,
.21201212412124521240B,21240212532125421302B,21330213502141321503B,
.21561216442175122064B,22207223062242022534B,22567226072272322760B,
.23305233072325623202B,23122231262315723166B,23162231322311423115B,
.23076230452275522705B,22654225752245522326B,22232221342206122042B
./
 DATA (((ICLON(I,J,K),I=1,18),J=1,1),K=4,4) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
```

```
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICLON(I,J,K),I=1,18),J=2,2),K=4,4) /
.20076200742007320071B,20117201312014520167B,20157202102021520172B,
.20121201022013320172B,20121200402001017753B,20027200472005520102B,
.20222202642026520132B,20176201772017620166B,20155201472013020107B,
.20137201722024120314B,20332203772044020502B,20517205172045020402B,
.20273202032010720053B,20010200252005220112B,20141201372015620211B,
.20206201702013220111B,20054200442003620051B,20060200602002220025B
./
 DATA (((ICLON(I,J,K),I=1,18),J=3,3),K=4,4) /
.20226202162020720212B,20272203122031220304B,20341204162042120352B,
.20274202342023320272B,20220201562006320053B,20101201032012720143B,
.20367204322045720325B,20373203542034720336B,20315202762020420213B,
.20277203512043720537B,20637207152077721060B,21130211402106520747B,
.20626204672036120304B,20233202352027520337B,20415204602051420532B,
.20475204252033620305B,20251202352021320217B,20207201732015520150B
./
 DATA (((ICLON(I,J,K),I=1,18),J=4,4),K=4,4) /
.20373203722036620372B,20400204502044220422B,20505205602055120500B,
.20407204402046220373B,20306202352014620136B,20151201532015320174B,
.20510205312055420553B,20566205542054020514B,20504204502035620416B,
.20474205652066420771B,21131212712135521444B,21512215052142721303B,
.21157210212066720602B,20541205542060020620B,20655207412101321026B,
.20761206432056520523B,20462204262040520363B,20347203342032320311B
./
 DATA (((ICLON(I,J,K),I=1,18),J=5,5),K=4,4) /
.21231212512125421277B,21317213112124721244B,21265212722127521263B,
.21237211462102520707B,20604204672042020335B,20272202562026420257B,
.20672207472102521074B,21124211372112621141B,21166211762123321262B,
.21214212472133621453B,21575217262210722326B,22445225032247522417B,
.22452223262216622130B,22075220572204422074B,22104221412214522177B,
.22204221542213022061B,21767216602153321454B,21372213212126421251B
./
 DATA (((ICLON(I,J,K),I=1,18),J=6,6),K=4,4) /
.22263222432222322161B,22115220342176621754B,21757217442172521665B,
.21576214742140421265B,21145212322074020637B,20540204742045420465B,
.21132211462120721245B,21324214002144421514B,21551216072163321653B,
.21754220042206722156B,22277224432226022676B,22772230522312623143B,
.23317232672324623232B,23211232342326023261B,23267232642325623250B,
.23261232332321123131B,23053230172274222642B,22541224512230622317B
./
 DATA (((ICMON(I,J,K),I=1,18),J=1,1),K=1,1) /
.20523206132066020714B,20740207752103421077B,21207212302132621576B,
.22015221222223012235 1B,22206222042172421613B,21635216062151621340B,
.20716205772047220401B,20332177111677516030B,15513153571526515226B,
.15151150541467214513B,14402142661421314175B,14224142451436514644B,
.15465160261640116721B,17175175332015620550B,21063213712163121663B,
.21756220542211122231B,22223221062174321604B,21407212062104020655B
./
 DATA (((ICMON(I,J,K),I=1,18),J=2,2),K=1,1) /
.20516200652064020703B,20752207542100021064B,21170213452143721536B,
.21745221212221022235B,22103220202173221652B,21630215732147621317B,
.20763206162045120330B,20216176771700316045B,15543153741530415221B,
```

```
.15157150531462714404B,14304141641413614053B,14101141641434314631B,
.15474160031631716606B,17235175622016520472B,21011213222155221670B,
.21754220252211622165B,22132220422172721570B,21375212242076520614B
./
 DATA (((ICMON(I,J,K),I=1,18),J=3,3),K=1,1) /
.20621205612060520643B,20730210112112221177B,21243213602157722002B,
.22015217642201621773B,21717216412164721622B,21623215522143521321B,
.20707205662037220201B,17730174231666416166B,15665154621533215226B,
.15140150161452014321B,14204140201374413733B,14026141721441114764B,
.15453157501612416447B,17040175132016020433B,20707211442137221534B,
.21653220062213322050B,22035217542166521551B,21427212752104220766B
./
 DATA (((ICMON(I,J,K),I=1,18),J=4,4),K=1,1) /
.20603205352056520644B,20764210552113721124B,21251213742164422034B,
.22103217712204122015B,21731216212160321622B,21606214732141421275B,
.20641204632024017730B,17474171471661216235B,15770155731542515302B,
.15151150021445214274B,14075137141364113672B,13772142101444415041B,
.15406155521604516432B,17040174452003220341B,20615210542123321350B,
.21527216442170421751B,21706216642162221514B,21417212602106321035B
./
 DATA (((ICMON(I,J,K),I=1,18),J=5,5),K=1,1) /
.20477205162054520567B,20631210032110021241B,21443215422166021747B,
.22062221302177121631B,21650216062142621355B,21274211632102720632B,
.20623203612010517615B,17270170511663316422B,16202160251562015436B,
.15226147371434414244B,14215141511376313701B,13725141151441614665B,
.15233155651613616423B,16651172151757720132B,20354205352066720762B,
.21121212072121121315B,21345213232131121222B,21134210702076020750B
./
 DATA (((ICMON(I,J,K),I=1,18),J=6,6),K=1,1) /
.20300201562011420165B,20446206032074021072B,21151212272136621262B,
.21631216722163621610B,21534214332136221274B,21177210762072320534B,
.20474202321774717475B,17262170751665716432B,16171157671554315363B,
.15171147551460314510B,14467144561443714457B,14560146601501015241B,
.15441157031621516600B,17101173461753517755B,20154202422026520407B,
.20517206122065020712B,20722207302074220725B,20640206112056620536B
./
 DATA (((ICMON(I,J,K),I=1,18),J=1,1),K=2,2) /
.20706206202070121027B,21132212072130721413B,21430214622145121504B,
.21504215762160121604B,21536215472153321561B,21607216352164321571B,
.21276211242100421342B,20732200461674616006B,15305146641443414536B,
.14516144251434414235B,14175141151406614001B,14120143561442214734B,
.15252156011621216704B,17213174632005120403B,21052214432177322203B,
.22222222172222122276B,22271222012204221667B,21537214022121421043B
./
 DATA (((ICMON(I,J,K),I=1,18),J=2,2),K=2,2) /
.20725206742075521034B,21114211652126321454B,21577215532160121605B,
.21565215122150121525B,21526215302153421564B,21614216322161221476B,
.21260210622076021277B,20641177021671616075B,15331146751472314631B,
.14556144151427714140B,14115140541401213775B,14122143341446014740B,
.15216156211616216607B,17112174562001420365B,20777214302175322142B,
.22213222302226122245B,22274221752207521715B,21544213722121621067B
./
 DATA (((ICMON(I,J,K),I=1,18),J=3,3),K=2,2) /
.20776207132076421051B,21135212252135221517B,21645217072204522033B,
.21727215612147621431B,21472215742161421651B,21653216172152521416B,
.21112207242055721170B,20327174261664216153B,15414150271506714772B,
.14652144521427214120B,14041137741375013767B,14123143061452014750B,
.15167156111612216372B,16653172751774420344B,20733213442165622017B,
```

```
.22052221242225622237B,22160221302202721700B,21540213552124721173B
./
 DATA (((ICMON(I,J,K),I=1,18),J=4,4),K=2,2) /
.21013207572077021051B,21213213262144121525B,21617217362207022145B,
.22030216242155721455B,21424215252162721630B,21615215412144221304B,
.20777205072023220755B,20031173061660616160B,15516151601523015071B,
.14726144631417014043B,14010137311374013756B,14066142341447115112B,
.15164156061604316216B,16532171231754620217B,20575210612142621627B,
.21744217342215722206B,22130220542177321640B,21503213442126321217B
./
 DATA (((ICMON(I,J,K),I=1,18),J=5,5),K=2,2) /
.21017210342107421071B,21102211442120621403B,21556217622203322062B,
.22017217372167221560B,21512214002137121334B,21261211542105120676B,
.20720203742011217566B,17254170201652716276B,16021156051543115256B,
.15051145541431414213B,14165142151422114220B,14237143471453214777B,
.15155155171600216245B,16502170641743317720B,20217203752054620773B,
.21136212622132421433B,21477215032142421304B,21252212432124121257B
./
 DATA (((ICMON(I,J,K),I=1,18),J=6,6),K=2,2) /
.20473205452054120605B,20707207712104521204B,21317214352157121653B,
.21705216432155321476B,21443213612127421206B,21056207262056020356B,
.20533202642004417603B,17317170471661716361B,16126156551542015213B,
.15075147471462414541B,14512144771452714623B,15010151531527215466B,
.15506157531622716522B,16764172221744617624B,17727201472030320370B,
.20512206042064720725B,20744207612101221023B,21016210112102121047B
./
 DATA (((ICMON(I,J,K),I=1,18),J=1,1),K=3,3) /
.21273213312145221570B,21701217622177321770B,21763220142203321767B,
.21726217202171721673B,21646216452161021564B,21574216372163321534B,
.21136206772047320206B,17536171101637315647B,15256150011455214436B,
.14340141411372513744B,13742137051371713773B,14135143741470115220B,
.15403157341627616733B,17227174511773720265B,20633212212162122030B,
.22155222232217322237B,22255221762207621721B,21576214572133321200B
./
 DATA (((ICMON(I,J,K),I=1,18),J=2,2),K=3,3) /
.21304213342144121551B,21653217152175522006B,22061220652211622117B,
.22075217542165721657B,21664216502160121547B,21535216172157721461B,
.21061205612037420226B,17464170211631515652B,15236150021462714536B,
.14400142331400713766B,13757137411375414027B,14163143621466115215B,
.15401157241630416675B,17107173651765720174B,20605212442161622023B,
.22131222542226022263B,22316222632213221765B,21552214152130021166B
./
 DATA (((ICMON(I,J,K),I=1,18),J=3,3),K=3,3) /
.21305213262143021547B,21642217122200122050B,22063221122222022201B,
.22107217612162121641B,21642215322162321564B,21576216212154721412B,
.20674203602010420313B,17325166761627515657B,15244150151473014607B,
.14500143221401313723B,13701136751373714061B,14170144141470215227B,
.15346157341624016474B,16657171551752720126B,20452210622146421740B,
.22124222522232722301B,22272221652207121727B,21553214132134021245B
./
 DATA (((ICMON(I,J,K),I=1,18),J=4,4),K=3,3) /
.21277213552142721536B,21674217732204122007B,22027221172220422235B,
.22135217622165321623B,21603215612154321526B,21533215342144121255B,
.20531201571761720031B,17171165601623115724B,15365151331507314746B,
.14546143401404613722B,13630136331373014071B,14242144711471315174B,
.15377157131612616274B,16503170361741020020B,20423207472132321620B,
.22036222452230022243B,22165221212200321702B,21563214412134221310B
./
```

```
DATA (((ICMON(I,J,K),I=1,18),J=5,5),K=3,3) /
.21223213202145021443B,21450215012154221634B,21764220462207622052B,
.22014217462165421606B,21476214212132521244B,21230211672104620702B,
.20333200101747517175B,16652163551606615704B,15547153541512714742B,
.14546143671421314117B,14122141401421314307B,14367145331475115154B,
.15323156311604416257B,16521170301732217636B,20147204222073721171B,
.21415215762170421741B,21651215372147721365B,21313213242133021337B
./
DATA (((ICMON(I,J,K),I=1,18),J=6,6),K=3,3) /
.20634207262072721111B,21161211732125221355B,21412214762157021601B,
.21621215452147721411B,21350213032122421123B,21011206542047620271B,
.20265200031753517274B,17037166021637416162B,15750155541535515147B,
.14772146331455314515B,14520145331461714705B,15045152351540515611B,
.15643160711633516575B,17016172141741317574B,17743201022025420426B,
.20606207002074121000B,21032210502105221061B,21074210722110021116B
./
DATA (((ICMON(I,J,K),I=1,18),J=1,1),K=4,4) /
.21511216052172722016B,22056220342175721707B,21643216242160521551B,
.21561216102165121610B,21642216342160121553B,21527215312145721417B,
.21011205672045620025B,17474170341631115550B,15200146371445014277B,
.14220141161374213620B,13572136111371314021B,14176144341474215306B,
.15562161141644417022B,17245174611766020077B,20321206102110021415B,
.21723221002225622344B,22374223422230522161B,22024216122146721321B
./
DATA (((ICMON(I,J,K),I=1,18),J=2,2),K=4,4) /
.21512216012172422014B,22012220122200321656B,21720216732167121643B,
.21620215742155121556B,21612216502162121575B,21560215422143021354B,
.20727205122036520071B,17422167711632215570B,15201146531445614376B,
.14310141411377713642B,13617136551373514044B,14211144601477315301B,
.15554160621642616734B,17124173331755117751B,20246205622105521404B,
.21676220572222222326B,22345223362224722116B,21774216202143521331B
./
DATA (((ICMON(I,J,K),I=1,18),J=3,3),K=4,4) /
.21513215662166621775B,21770217762201622013B,21744217372176221725B,
.21633216062151621520B,21576216422160721570B,21537214512137021237B,
.20610203302012520253B,17371167471631715620B,15227147351455714473B,
.14347142131406213704B,13577136021370714035B,14213144521474115276B,
.15545160531634716607B,16735171411742017723B,20167204622101421357B,
.21650220532220622272B,22315222622216422043B,21734215722145621322B
./
DATA (((ICMON(I,J,K),I=1,18),J=4,4),K=4,4) /
.21473215642166522014B,22125221212205722007B,21751220012177721755B,
.21614215332152521506B,21522216012157021510B,21372213402124621147B,
.20504201411762517650B,17277166571627315705B,15300150051467414544B,
.14376142221405413720B,13607135711370314035B,14225144461470115231B,
.15537160301627016436B,16615170441731317621B,20125204072071421255B,
.21564220112214322232B,22252222222210722010B,21701215652144021340B
./
DATA (((ICMON(I,J,K),I=1,18),J=5,5),K=4,4) /
.21461215352156121604B,21626216432165221725B,21762220132200221760B,
.22003217702173321663B,21575215242144521370B,21313212672122721164B,
.21211207742050720134B,17510170741650616125B,15524151531456714335B,
.14073140431375013671B,13654137321374314073B,14060142561447615001B,
.15260155761600616172B,16407165071702517315B,17556200742033320642B,
.21041213022146421600B,21667216312154321545B,21527215152151221521B
./
DATA (((ICMON(I,J,K),I=1,18),J=6,6),K=4,4) /
.21[illegible]211652126721353B,21374213732141321462B,21512215272153221556B,
```

```
.21574215522153321513B,21433213462126021172B,21115210222072420632B,
.20534203272011017656B,17345167471645116146B,16004154231520715033B,
.14666145461444414375B,14423143721442314514B,14610147661516415414B,
.15566160231620516443B,16637170271722117410B,17560177542012520257B,
.20421205612066320760B,21062211602120221224B,21277213222132621361B
./
 DATA (((ICNON(I,J,K),I=1,18),J=1,1),K=1,1) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICNON(I,J,K),I=1,18),J=2,2),K=1,1) /
.20215201742017120222B,20214201672021720265B,20307202462031320332B,
.20307202572017620056B,17774177221760617502B,17536176041763717706B,
.17636177101772617762B,20015200462006520074B,20115201242013120126B,
.20107200572003620151B,20127201132013520134B,20062200542004717766B,
.17723200022002320050B,20051201072007420013B,17745177421767717711B,
.17660176711776020037B,20035200512006320037B,20036177772006120143B
./
 DATA (((ICNON(I,J,K),I=1,18),J=3,3),K=1,1) /
.20406204152040520430B,20432204502047020520B,20554205342056520567B,
.20605204732030120204B,20120177751770717633B,17633177241767317704B,
.17702177231774017770B,20054201412022320245B,20255202602025520242B,
.20203201342014020151B,20113200772012420156B,20143201062006617772B,
.17737200742020020221B,20243202342017020052B,17777177421770717667B,
.17673177312003020047B,20105201232015520160B,20153201722027120345B
./
 DATA (((ICNON(I,J,K),I=1,18),J=4,4),K=1,1) /
.20611206132060320634B,20647206672071020750B,21003207532100421006B,
.20702205672046120334B,20274201372003620007B,17776200012000020007B,
.17765200022001620055B,20146202412033620400B,20414204102040120342B,
.20275202242020720167B,20134201352016720160B,20202201752013220073B,
.20105202502033120334B,20353203442017320104B,20052200422001417774B,
.17770200372011320161B,20220202512027020310B,20333203502043220536B
./
 DATA (((ICNON(I,J,K),I=1,18),J=5,5),K=1,1) /
.21420214512145621501B,21474214712144321441B,21460214762150721417B,
.21301211752107220747B,20652206372061120571B,20545204712047120462B,
.20411204052041320414B,20431204732050420500B,20465204522046020470B,
.20501204142035320321B,20304202702025120261B,20307203442045220540B,
.20411204612054220525B,20464204742045420445B,20443204532045620455B,
.20530205672061220650B,20730210032105721123B,21132211762131121377B
./
 DATA (((ICNON(I,J,K),I=1,18),J=6,6),K=1,1) /
.22071221552220522106B,22062220002200321667B,21703217312173221657B,
.21571214722143221375B,21342213042126121262B,21267212712124521223B,
.20725207542076720750B,20733207052066520624B,20576205442051020464B,
.20447204222041420415B,20415204262044420434B,20450205012055320611B,
.20540205532056720633B,20720207342075120776B,21024211012120721236B,
.21257212732132621351B,21412214562152021602B,21716217552200722042B
./
 DATA (((ICNON(I,J,K),I=1,18),J=1,1),K=2,2) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
```

```
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICNON(I,J,K),I=1,18),J=2,2),K=2,2) /
.20210202142020520221B,20236202102017520205B,20221202562022320232B,
.20207202052013120106B,17767177071766117637B,17663177021771317751B,
.17677177461777620036B,20102201242011620115B,20104200672020720175B,
.20153201042010120113B,20052200672007020013B,20014200222001617752B,
.17731200072006220073B,20100200612001717763B,17740177301772717723B,
.17715177461773120023B,20060201002007120070B,20055200612007620120B
./
 DATA (((ICNON(I,J,K),I=1,18),J=3,3),K=2,2) /
.20400204342044520427B,20461204372046020435B,20455204202042720376B,
.20363203262023220166B,20125200311776417730B,17755177221770617763B,
.17765200072004520116B,20164202422027520307B,20247202062041720364B,
.20274201472006320137B,20073201032006720033B,20012200172001617755B,
.17770201052023420235B,20221201252002520001B,17753177411771517675B,
.17706177512005620123B,20152202032021520215B,20216202322025620321B
./
 DATA (((ICNON(I,J,K),I=1,18),J=4,4),K=2,2) /
.20614206452065420675B,20664206632063120646B,20676206552063420613B,
.20531204552041220317B,20225201612007020046B,20045200432003720071B,
.20134201332013620200B,20260203502041520453B,20441204122054720504B,
.20355202302013120125B,20130201212011120103B,20065200532006120075B,
.20045202662036420347B,20276201732014620114B,20063200322001117767B,
.17775200362015420245B,20303203362036720375B,20405204262045020511B
./
 DATA (((ICNON(I,J,K),I=1,18),J=5,5),K=2,2) /
.21427214432143021441B,21515215002141321443B,21427213762137521333B,
.21236211772110421011B,20727207002065620642B,20646206372064220645B,
.20557205432054520565B,20570206212065520650B,20637206332061520570B,
.20526204642036220346B,20327202732025620251B,20252202542032320461B,
.20506205542056120543B,20557205552054120540B,20541205362052720541B,
.20560206202066320717B,20770210412107721137B,21166212272130121352B
./
 DATA (((ICNON(I,J,K),I=1,18),J=6,6),K=2,2) /
.22055220652206622034B,22057220332200321716B,21710217252174021666B,
.21621215442147321450B,21455214072137621377B,21371213602133121315B,
.21134210612103621031B,21045210242077620753B,20757207132062720613B,
.20576205702055120526B,20516205152047120455B,20513205412056020612B,
.20655207242076521011B,21052210642106721100B,21127211642122221311B,
.21354213762142321450B,21463215072154121566B,21665217132175421765B
./
 DATA (((ICNON(I,J,K),I=1,18),J=1,1),K=3,3) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICNON(I,J,K),I=1,18),J=2,2),K=3,3) /
.20241202352023520230B,20232202122016420157B,20210202012020620234B,
.20247202522024020225B,20225201312007720055B,20040177511772017735B,
.20034200252005220074B,20110201442016720226B,20235201632023620225B,
.20173201562010420051B,20023200162000717772B,17762177641777017773B,
.17754200322006020101B,20121201332013520075B,20030200142002420027B,
```

```
.20022200122004220131B,20132201212012720121B,20115201042011320126B
./
 DATA (((ICNON(I,J,K),I=1,18),J=3,3),K=3,3) /
.20437204442044320424B,20414204012036720353B,20402203552036620365B,
.20400204142036420365B,20354202512015420111B,20124200532001520014B,
.20077201032012320146B,20230203222037220450B,20474204322047220416B,
.20350202722014420115B,20062200612003220005B,17770177722001420027B,
.20000201672023020256B,20277202632025120203B,20123201012006720054B,
.20071201032016520242B,20302203022030520306B,20301203002030420317B
./
 DATA (((ICNON(I,J,K),I=1,18),J=4,4),K=3,3) /
.20673206672067220661B,20640206142056720605B,20605206052057620575B,
.20563205542052620500B,20453203422027120215B,20176201432013320151B,
.20217202212025120271B,20367204512052720570B,20642206602063020557B,
.20475203772026520170B,20146201212007520052B,20045200542011320102B,
.20164203172041520424B,20440204102032120306B,20241202022017120174B,
.20212202532032520370B,20424204542047520500B,20473204722047420476B
./
 DATA (((ICNON(I,J,K),I=1,18),J=5,5),K=3,3) /
.21456214572143321447B,21462214172136421347B,21335213162130321262B,
.21232212052115121107B,21044210142077320754B,20734207232071020700B,
.20664206422065320660B,20722207542075520770B,20772207752077120743B,
.20653205672052120444B,20407203642034120315B,20311202572030620407B,
.20577206342065420643B,20646206422063420633B,20645206652071420740B,
.20755207762102021035B,21103211372116121173B,21210212152123421260B
./
 DATA (((ICNON(I,J,K),I=1,18),J=6,6),K=3,3) /
.22061220542203221754B,22002220102176121744B,21721216662166021565B,
.21556215072145221405B,21423214102135521337B,21326213172127721264B,
.21240211722114421126B,21116210752106021046B,21020210052076420742B,
.20721207022065320620B,20566205262051020465B,20456205142055620617B,
.20734207652100521015B,21020210412106121076B,21130211602123321277B,
.21346213742141021425B,21420214302145321473B,21507215732161321625B
./
 DATA (((ICNON(I,J,K),I=1,18),J=1,1),K=4,4) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICNON(I,J,K),I=1,18),J=2,2),K=4,4) /
.20235202332023220231B,20200201662017320160B,20166201632020320223B,
.20237202532024420174B,20177201462011020070B,20054200362003320034B,
.20056200632010020120B,20146201742021420231B,20216202022014620204B,
.20171201432012120106B,20103200532004120025B,20024200122000720022B,
.20034200452007520112B,20116201272014120125B,20113201002006220062B,
.20066201012013520151B,20157201672017520171B,20167201612015120164B
./
 DATA (((ICNON(I,J,K),I=1,18),J=3,3),K=4,4) /
.20467204662045120423B,20401203622034420355B,20367203612037020413B,
.20425204232040020364B,20374203512026620210B,20212202012014720102B,
.20145201622020720225B,20301203562041120446B,20440204122040520367B,
.20326202662022420171B,20135201242010720064B,20046200362003320037B,
.20117201752021720242B,20272202752027620265B,20240201772015120140B,
.20151202072026620325B,20347203702037420373B,20374203672036520370B
./
```

```
 DATA (((ICNON(I,J,K),I=1,18),J=4,4),K=4,4) /
.20722207122066420640B,20611206012056620554B,20545205512055120561B,
.20545205112053520516B,20541205062041120334B,20347203432030020243B,
.20257202742031520362B,20455205322056720621B,20624206172053720502B,
.20445204162035420322B,20271202232016320127B,20112201122012420143B,
.20227203012036120367B,20405204132041320373B,20344203142026420237B,
.20255203332041220463B,20520205452056520564B,20564205712056220560B
./
 DATA (((ICNON(I,J,K),I=1,18),J=5,5),K=4,4) /
.21455214512143521422B,21375213412131321305B,21266212312121121216B,
.21225212062116421150B,21112210752105021030B,20771207222070220704B,
.20653206662071020747B,21010210322103121036B,21033210042073320676B,
.20604206162053520500B,20454204302041520407B,20406204222044320476B,
.20655207032073220754B,20746207462075220767B,20776210032100121035B,
.21044211002112321154B,21223212462131021331B,21353213532136521366B
./
 DATA (((ICNON(I,J,K),I=1,18),J=6,6),K=4,4) /
.22103220742203722010B,22033220202176721756B,21716216572163021635B,
.21576215552152521524B,21503214522144021422B,21375213342131021274B,
.21227212352123621240B,21232212302121621203B,21200211722114421105B,
.21047210072075220722B,20665206412062420605B,20577206122063020652B,
.21030210632112121140B,21144211552116721210B,21241212752134621414B,
.21463215362156321613B,21635216512166021675B,21706217172177221757B
./
C
C
 DATA (((ICXOFF (I,J,K),I=1,18),J=1,1),K=1,1) /
.06036064010711707777B,10551114731244213427B,14062147371562116237B,
.16533167431722420014B,21100213752162021545B,22221226772334224176B,
.25501260622632527065B,27553302243044230574B,30506303633035630144B,
.27451267202622126031B,25660257122551025143B,24527244242375623176B,
.22522230632222421340B,20300167401602715151B,14200132171244111773B,
.11270106261034310035B,07436071040673006656B,06622067160666506703B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=2,2),K=1,1) /
.06020063600704507553B,10347112571215312646B,13511141051500415614B,
.16130163731674117266B,17502200312012320432B,20475213012232723107B,
.25100256222636027054B,27566300333021130330B,30441303363017127733B,
.27232265042617125750B,25057256042544125147B,24571244012373723157B,
.23664225342210421172B,20027165231542414444B,13564130101226711574B,
.10760102751012607575B,07135067130664206367B,06376063050640306475B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=3,3),K=1,1) /
.04766052770566106355B,07307102301115511777B,12627131331367014361B,
.15211155161633516734B,17331175771777320301B,21014217032272323264B,
.25263260562657127246B,27550301663027530407B,30501304523026527772B,
.26020267732655126337B,26155263152602025426B,25016244552401323231B,
.23417226562206020771B,17642165611546114524B,13560127211213011422B,
.10637100330765307511B,07173066400640606301B,06323060550616706345B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=4,4),K=1,1) /
.04524047660543405527B,06364071470747410364B,11015115021224113130B,
.13775143621550616210B,16676174171732517461B,21020213622205722675B,
.25141260402653327243B,27643301523031430447B,30512304133020127626B,
.27445272402711426636B,26405262742635726071B,25310245052424023435B,
.23404227722171520623B,17516164651551114465B,13544126241174311241B,
.10576103170775107572B,07520071020624606412B,06070060110621506475B
./
```

```
 DATA (((ICXOFF (I,J,K),I=1,18),J=5,5),K=1,1) /
.04576047750524005603B,06370070410760310107B,10461111371205012765B,
.13301144071532216334B,17054174722030521265B,21406212512241623225B,
.25330263012664127231B,27522300203023130377B,30503303503027630135B,
.30141301523013730103B,30012300522760727507B,27167263212550024705B,
.24705236612260521433B,20161170431561215001B,14133133751265112326B,
.11656115171114510464B,10110077140754707637B,07525075100757307740B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=6,6),K=1,1) /
.07467074320765310003B,10563103001105111501B,11655124531335013377B,
.14453151551617516775B,17262201232051721171B,21633222272252223016B,
.24245243512477325417B,26042264342663727045B,27224273012742427330B,
.27326273372731327400B,27335273342723427057B,26466257022515224537B,
.23773231152225621452B,20765201201730116555B,16061154601527514620B,
.14023134001273312427B,12042115521141511326B,11115113601154711572B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=1,1),K=2,2) /
.03775043370506505770B,06514073531032511354B,12641137601474415654B,
.16271166631717317672B,20442207332076221104B,21542222312306324136B,
.25515263662712227762B,30503312103140531551B,31622315763146331246B,
.30424276222677626605B,2660265232633126122B,25536247742435123422B,
.24104232142233421432B,20357170251556514463B,13422124231164010753B,
.10240676270716006673B,06127055410531505220B,05014052570525205564B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=2,2),K=2,2) /
.03561041760462505357B,06320071750774611062B,11755131171406515065B,
.15516160501652217166B,17624202302037720561B,21131214322265623551B,
.25621265022734330235B,30700312463153631733B,32014317503161331223B,
.30530277672727727137B,27102266602650626145B,25426247312426123463B,
.24250232312241421367B,20445167001522114115B,13034122471143511115B,
.10310076520723306730B,06167055460532105053B,05111051030532205412B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=3,3),K=2,2) /
.04171043760513105614B,06405074201021411035B,12034127661361214614B,
.15562163561664417442B,20102204212066121050B,21463225442341324261B,
.26035266072751630362B,30771313233151231547B,32106320103154331164B,
.30607302232761327354B,27267270352652526476B,25633250752433623613B,
.24356234272242521372B,20025167431553214365B,13345124051146311003B,
.10661075310707707054B,06216057760546305233B,05301052570536605412B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=4,4),K=2,2) /
.03707042220453405232B,06424072011020411162B,11741126541365414500B,
.15340160221661117341B,20021203452065321161B,2172226332355424155B,
.25705266622747130265B,31031313623156532001B,31757316443172731065B,
.305743064763013527650B,27315272412677526523B,26140253532464324112B,
.24425235742251221321B,17764166371557314545B,13540124331140710520B,
.07771074660730306767B,06324060630553305334B,05257052370532105475B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=5,5),K=2,2) /
.04433050710544506001B,06733074251025211053B,11701125001342214060B,
.14644156051660017444B,20211210152166322711B,23071235302433225011B,
.25535264112731730070B,30544310543123231416B,31504314563125131017B,
.30671307043061630250B,30131277752756027254B,27100265642572425007B,
.24004240562267221555B,20365170521566014624B,13723131521230211424B,
.11113107321035607655B,07261067000654706243B,06062060650626706654B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=6,6),K=2,2) /
.07055071610732007167B,10127104761126012017B,12275134141426014777B,
```

```
.15540163571723020153B,20442207352150521771B,22734235112430524756B,
.24236247562536426115B,26035270612731627502B,27644300263007130155B,
.27734276062752527446B,27461272372700326442B,26071254632476624263B,
.24557235042246121436B,20525176651707616370B,15537146511430713703B,
.13347127111212211525B,11162100431041610352B,10375101301047010614B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=1,1),K=3,3) /
.02227025430314503547B,04740056450674410126B,11212120531310014153B,
.15114156421616117001B,17527205202121021674B,22133226362364324347B,
.26510275503051231456B,32053330323331433513B,33560334633335732551B,
.32225316023123030572B,30230300442760627202B,26516257052500024065B,
.24370235422255221537B,20477172351607614612B,13417122221106307764B,
.07127063230567605315B,04641043610411404021B,04017042350450105040B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=2,2),K=3,3) /
.02332026570311703762B,05020057250700410051B,11103120351275013676B,
.14662155161627517165B,17710205622132121765B,22304230352374624441B,
.26625276003047731473B,32257330523341733563B,33550332223330732666B,
.32271314063123130500B,30131277112743427067B,26346255712474724016B,
.24404234722254521536B,20436172151573414370B,13067120071075710057B,
.07125063230566405306B,04655042770415304022B,04011042510452605005B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=3,3),K=3,3) /
.03320036010413104544B,05467064440736310420B,11223121321307313753B,
.14667154721665417666B,20372210212154122215B,22550234662427625123B,
.26543276613032531510B,32326327743327733530B,33552335333333433022B,
.32413320273133030746B,30346277762753227106B,26377256552473724045B,
.24637237032244721504B,20305170521565114376B,13177120201074010013B,
.07036063150566605350B,04731044140422604204B,04213044170465005153B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=4,4),K=3,3) /
.03545040140435605027B,05654066330747010513B,11520122711324414140B,
.15035160051704417716B,20607213132217222625B,23412243022507625511B,
.26476276713045531360B,32234326363326433522B,33564335273327733063B,
.32520321233154731155B,30654302252767227223B,26464256672511124376B,
.24776237412260121511B,20214170171556214432B,13274121231076707751B,
.07214065330602305451B,05073042730436104231B,04314044760503605321B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=5,5),K=3,3) /
.04736051330537406073B,06621073521034111255B,12107127541364014465B,
.15425163451733620320B,21132217542270723556B,24330246252543626064B,
.26220272413015530751B,31537321523240432627B,33021327243257432344B,
.32035316633154031221B,30643277612737027134B,26524256122474725344B,
.24135227453031421624B,20445171401570414506B,13401124231155411002B,
.10264076260726006673B,06403062100603605456B,05472055620611106346B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=6,6),K=3,3) /
.07221073240754307742B,10402110741156212255B,13104137571454315345B,
.16154170131765120600B,21315217662243023165B,23671244522524626024B,
.24644254642623126724B,27417277403024130432B,30541306233071530703B,
.30664305133025330005B,27516272472675026423B,26035254102473224146B,
.25101237142257421534B,20574177361701416131B,15212144211375713241B,
.12623122011155611175B,10644103551010607667B,07570077511014310245B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=1,1),K=4,4) /
.02542027230321304056B,05211063500733410426B,11442123611327314161B,
.15025156171626017024B,20142212352211222754B,23607242312532426027B,
.26652276453061331560B,32445332173357534016B,34115341123400433534B,
```

```
.33224327363240332005B,31302306503030127616B,27132262302530424333B,
.24611235622255421517B,20536174021625315166B,14002125661135310065B,
.06707057340521504571B,04135035100320103055B,03062031650343603736B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=2,2),K=4,4) /
.02547027120330104213B,05450065340754010413B,11441124211322714065B,
.14750155451630617301B,20166212022211722754B,23522243442517326005B,
.26672276133064531564B,32420332113362734034B,34131340673354533511B,
.33217327373237431777B,31276300653030727637B,27117262232535624415B,
.24614235732267221546B,20462174121623415026B,13531124361107507757B,
.06744060100525104534B,04224036150326103131B,03057031240337703763B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=3,3),K=4,4) /
.02737031660353104322B,05530064340741610416B,11413123011315414027B,
.14720156311645517560B,20423212522226023113B,23660245162533526036B,
.26712277353070731630B,32412331163351734024B,34064337603361133501B,
.33324327533234031750B,31403310543041427653B,27062262212537724535B,
.24753236742255521521B,20412172531612314765B,13543123311102607732B,
.06721060160522604525B,04115035630337103222B,03127032560350304154B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=4,4),K=4,4) /
.03036033210362104357B,05330063320745110465B,11363122311307213713B,
.14715160031660517531B,20524214322231323245B,24057250222546626243B,
.26665276423065631464B,32277330213334533655B,33727337373366233511B,
.33252330453244632022B,31462312053055727756B,27170264102563024761B,
.25062237422265121553B,20414172461601214741B,13512124241105210021B,
.07127060770542404572B,04141035740345203264B,03277034450373404325B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=5,5),K=4,4) /
.04027042300457505263B,06046067740751010432B,11274121551305013727B,
.14726157141670017673B,20670216352252623527B,24345250132544126234B,
.26426271763023430726B,31611322543257433037B,33125331503315633047B,
.32657324613220731755B,31465311003043330050B,27421266502574525025B,
.25207241202307121774B,20570174451623115024B,13672127401162511045B,
.10132074270664506145B,05540052400510604710B,04713047630512205406B
./
 DATA (((ICXOFF (I,J,K),I=1,18),J=6,6),K=4,4) /
.06411064160654607017B,07456102151075411507B,12270131351377414632B,
.15467162561716720047B,20645214612231023104B,23534242202467425364B,
.25064256262642127072B,27562301513053331012B,31157312563134331323B,
.31236310713072630545B,30265276752735026737B,26274255632477724045B,
.24603235162260521642B,20726177651702416111B,15261143401357313037B,
.12244114431102410446B,10135076010732107172B,07131071310730107473B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=1,1),K=1,1) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=2,2),K=1,1) /
.20743206012041220632B,20652206312101121016B,20565203062016417766B,
.17565175441737517214B,17115167271654416075B,20301201771655216110B,
.17601201641760517524B,17606175541754017550B,17626176101766317543B,
.17751176441764717427B,17354175102011720343B,17761206172050220503B,
.21631212232103421012B,21027211072072520464B,20316201612006620233B,
```

```
.20226177221761720007B,20153202672036417747B,20157176021772320057B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=3,3),K=1,1) /
.21137214342160021760B,22141222002226722123B,21405213022120220661B,
.20635202341772517402B,17303172111672516513B,17151166421657116613B,
.20653213402111520577B,20624207472117621162B,21212211022063120443B,
.20644205552063120551B,20633210312115322155B,22215222252217522231B,
.22676223502223522316B,22346220552161421352B,21263211042065020525B,
.20362202622036020567B,20574207322063220600B,20704202042057520716B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=4,4),K=1,1) /
.22321225072304023240B,23214231542316023115B,22417222262204721533B,
.21213211452074420510B,20164201251756217232B,17506176311756417517B,
.21506215762150521215B,21166211462162221705B,21643215412150022071B,
.21605215452103521260B,21306216742205122445B,23203233452311023042B,
.23432233752332723316B,23155231022301622560B,22327221722167121520B,
.21363212512120721241B,21243215462157121526B,21551214052163222031B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=5,5),K=1,1) /
.25704260152633426461B,26540266012652426374B,26460265762641726057B,
.25437247532443224172B,23607234222343223633B,23560234642357524047B,
.25060250542474524725B,25000251242530625437B,25463254062523125223B,
.25111250142476624663B,24706250502517125330B,25550256332611526313B,
.26230264342616226117B,26173260572601125667B,25564256042561025615B,
.25627257122567725552B,25444253722534625522B,25316253402535325444B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=6,6),K=1,1) /
.31611320773217232007B,31647315173152631422B,31305313243141131157B,
.30715305473034530135B,27640273642720127305B,27132271012667726716B,
.30041301523017030215B,30115277702760727675B,27623275152746727640B,
.27725277423002630116B,30161303303043230634B,31004305303064731121B,
.30247310253045031012B,31074310773102630743B,31110314603171432064B,
.32005321203204331672B,31674316123147431513B,31314313473157731556B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=1,1),K=2,2) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=2,2),K=2.2) /
.21024207032077020762B,21052211202102121013B,20770204362024120113B,
.17760177321764317130B,17063170621670016642B,17114205171664617071B,
.20260203122037420454B,20475205502045320555B,20613204562026420161B,
.20107201752006720002B,17741201252034220505B,20624206742056520567B,
.21126207132053620564B,20403204362021120070B,20026177562016120316B,
.20056175001752217760B,20023201022010017551B,17665177552011520003B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=3,3),K=2,2) /
.22141222252237122301B,22524225312253622335B,22005214552132620773B,
.20565203612004617554B,17420173351746317335B,17533174251751117654B,
.21172216522165221661B,21615217022175521663B,21561214212117421062B,
.21162211302076020754B,20740211072131222103B,21715216072140621653B,
.22405222332211022023B,21673214772123721043B,20740206242051120363B,
.20141201432031520540B,20704207732070420655B,20664207672102521071B
./
```

```
 DATA (((ICYOFF (I,J,K),I=1,18),J=4,4),K=2,2) /
.23375235222377024070B,25152240452360523570B,23066226352245122060B,
.21610214172123020734B,20522203372025720214B,20273206142074621037B,
.22606227572302323014B,23053232012327023211B,23052225472272122370B,
.22344223012217221757B,22011221652244722601B,22564226312252622266B,
.24047237372361423322B,22763227032265322421B,22256220702155021307B,
.21232210512125121523B,21644220102205222116B,22170221622227122313B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=5,5),K=2,2) /
.27560306313067030713B,27713300032767727506B,27421273332716427024B,
.26413260712564425313B,24720244322553025615B,26112261762634025435B,
.27132271262713227144B,27354275212760027510B,27366273522717226723B,
.26433261232577225775B,26016261262627326365B,26366264672627326705B,
.27205302772712327233B,27473275162712026631B,26632265562655726614B,
.26364263722642026305B,26173261652611726000B,26026262012642326361B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=6,6),K=2,2) /
.33507334733350233353B,33454333173321333221B,33034331023316432622B,
.32435322273204231626B,31456314303130231121B,31026310633106031060B,
.32422325373255032401B,32212320133167131601B,31450314723143231550B,
.31422313663132631377B,31575316443165431661B,31657317003205232356B,
.32232323673245232462B,32433323523240032402B,32441325073253432746B,
.33136330133307433107B,33053327553261632534B,32501325433271032501B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=1,1),K=3,3) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=2,2),K=3,3) /
.21217213262133221271B,21337213262133721366B,21360212742126721275B,
.21055207462064320552B,20652204072023620362B,20273202562040417666B,
.20702210752115621242B,21246213072131521170B,21212212332102321043B,
.21015210022103320747B,20714207332101421173B,21137210242063120450B,
.21364212142115221154B,21165212512130321253B,21021206352055620524B,
.20475203222025220343B,20444204652043220366B,20335204172044520415B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=3,3),K=3,3) /
.23400227722305123115B,23235232632325723051B,23035226232245322366B,
.22306222412173621521B,21434211002047320602B,21165211732105420615B,
.22160222662263222772B,22740230342314623114B,23056227642253422400B,
.22427225012232122273B,22250222372234622461B,22431222512165121374B,
.22716226242247622412B,22447225462265122533B,22336220562164221600B,
.21457213642120621302B,21413214752140721431B,21403213732144021464B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=4,4),K=3,3) /
.25062252052456724670B,24703247702473524572B,24416242512401023722B,
.23641235112332523012B,22534221652172521533B,21715216622164421727B,
.23463236142412024501B,24513246142500224756B,24707245532427224140B,
.24115241502370223605B,23635237662411524046B,23621232762275622476B,
.24510243612422624147B,24167242672411624004B,23747235632332023177B,
.23114230322275222534B,22603225752261422601B,22601225722270622747B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=5,5),K=3,3) /
.31016310313107431227B,31472313663114730770B,30627306443040130116B,
```

```
.27704275242725026705B,26370261572612326064B,26016257062565526021B,
.30001300033015730316B,30700310123112331204B,31140307723052130344B,
.30133277542767027454B,27370271712716327164B,27051264012700631074B,
.31053304132725630333B,30615306563043330331B,30304303103032530317B,
.30227301603004227604B,27453273312721026771B,26713267052667026726B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=6,6),K=3,3) /
.34725345373442534253B,34263342333416134111B,33662335313351633342B,
.33144327253241332051B,31621314503126731105B,31060311243120631211B,
.33342333133324733274B,33235331473310633052B,33042327703271732624B,
.32600325433243032206B,32001316723174131741B,31640316273175732245B,
.33706335503347333505B,33430333753334633370B,33421334553357633753B,
.34112337763364733562B,33467333433222233062B,32643327503263032627B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=1,1),K=4,4) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=2,2),K=4,4) /
.21441214412141421404B,21365213242136621405B,21415214572151421472B,
.21344212642125321130B,21066210072063420545B,20703207252070320646B,
.21065211772125421336B,21410214432143021372B,21422214052133721315B,
.21333214732156121600B,21570215232155521513B,21465212712111621000B,
.21320212222117721120B,21053212152132221323B,21320213012125421113B,
.21016207452077121005B,20711207312076021044B,21072210042071620721B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=3,3),K=4,4) /
.23426234252346023475B,23136230452302323050B,23104231142312223033B,
.22607230302241422223B,22143220602165221560B,21616216102151221210B,
.22352225742301023131B,23151232352325123246B,23176231112300123015B,
.23055231262327723331B,23263232202332522316B,23014226112243522213B,
.22744227272255522505B,22473226172273023001B,23024227202257122424B,
.22304221472223022175B,22130221122213422214B,22233222622221522175B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=4,4),K=4,4) /
.25101251542520725201B,24601246212463024531B,24511244552441724340B,
.24140237622375423560B,23404232722304022640B,22760227622273722534B,
.23665240712450224046B,24707250272516025131B,24741246462462224562B,
.24610247032473524771B,24774247262470624572B,24425242332402323467B,
.24444244372430124130B,24151242212433624256B,24336243452414723724B,
.23536235452361423614B,23544235222352323455B,23512234502350523435B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=5,5),K=4,4) /
.31404314733156531660B,31716316033152431435B,31220310403071130546B,
.30443302433001327607B,27420272402721327230B,27205270662674626630B,
.30112303473053030744B,31216314343161231637B,31553315303142131236B,
.31055307103055430521B,30535305213043330373B,30312301702763027557B,
.31044310053060530650B,30724310333063430534B,30575306073106531075B,
.31232311343111231002B,30653305103040530264B,30177301013002327765B
./
 DATA (((ICYOFF (I,J,K),I=1,18),J=6,6),K=4,4) /
.35427354313530635250B,35233351363511535027B,34634343623421234157B,
.34045336703351133167B,33062327653257332410B,32163320433205031757B,
.33502335643370334015B,34101342413425334334B,34363343573427634161B,
```

```
.34062337773376633745B,33667333513316233075B,33026327113263232602B,
.34357343333442334357B,34412344133443534477B,34451346413477735142B,
.35106353403531135156B,34756345773437334201B,34030336723365533401B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=1,1),K=1,1) /
.23634227412202021016B,20141171131607515137B,14377135211310012600B,
.12510124261233312256B,12056120561174211654B,11535115421134711306B,
.10457106021102211131B,11443117141236413153B,14014147161555116451B,
.17316177612021420037B,17706176031770720134B,20476210202156122151B,
.22477230052317023060B,23216240102415024437B,24647246702475324757B,
.25021247512474024707B,24535244452430424175B,23724236332337423173B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=2,2),K=1,1) /
.23674227622205221204B,20243172361620515323B,14377137441324412757B,
.12641125731234712310B,12375124001237612166B,12310121531200711611B,
.10721107411073011065B,11530116771242113163B,13720145561555516437B,
.17331200172012517707B,17605175521766520174B,20472207672143322073B,
.22405227362317123342B,23606240432453325151B,25263251522514225170B,
.25210250222467524776B,24627245272435424174B,23773237452352023144B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=3,3),K=1,1) /
.24172233422246421557B,20541175171644015433B,14516141031344413063B,
.12715127631260712407B,12307123751235012250B,12335121621172211765B,
.11062110101105311220B,11477121701254313257B,13725146041563016515B,
.17723176522002120034B,20012202071773620244B,20550211042155322204B,
.22310226742317323543B,24042243462464125173B,25352253612532725270B,
.25203251372511124737B,24631245622442424241B,23770240702363323253B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=4,4),K=1,1) /
.24261235302252421745B,20741201041722116273B,15406146041410613434B,
.13255134001303312555B,12363123021240112404B,12204124341245312432B,
.11220111631125011430B,11635121531257513316B,14140150261573516427B,
.17202175321774020110B,20153200661774020165B,20566210772144322107B,
.22230224542320523715B,24312245262500025231B,25354254452546725360B,
.25264252032527225065B,24575245162452024212B,24150241332355723214B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=5,5),K=1,1) /
.24562240332332222524B,21626207201771517053B,16270156471507214314B,
.13424134401316712712B,12544125571250112371B,12447124571247612560B,
.11226114161166712171B,12400126701325214003B,14427152571606716553B,
.16752173201760320010B,20163200712017420351B,20521211162137221700B,
.21612222612272523553B,24266246012503725120B,25255253062536725341B,
.25307251772510724734B,24534243132416524057B,23770236542347223127B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=6,6),K=1,1) /
.23760236432325422601B,21744214042045517527B,17117164031567115475B,
.14727144231423514042B,13370135171352413527B,13561135111354713756B,
.12422127451307013326B,13606141701457615163B,15472160251641016760B,
.17300175551774217760B,20011200112004520157B,20361207422115021333B,
.21176220132221422662B,23155233612360523771B,24132241662420024170B,
.24236242112411223706B,23645235762343523254B,22731226222250222411B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=1,1),K=2,2) /
.24326235502264621676B,20650174641631515244B,14252131661232312076B,
.11721114361130011456B,11445114671145211400B,11354112161106610715B,
.07751100531020510330B,10645112631177612641B,13526144661555516617B,
.17560203372063320356B,17670177722014520377B,20736213022155422127B,
.22337230042321523063B,23252240442454225103B,25345254752547225610B,
```

```
.25745260462601525755B,25674254162513224677B,24372237772353623342B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=2,2),K=2,2) /
.24434235562264222012B,20631175511651115444B,14323134361256312242B,
.11760120001156111526B,11555116131154511462B,11514115071121311077B,
.10021100361014210244B,10613112621173212474B,13444143431546716500B,
.17402201022045420072B,20015177652010520423B,20734212542161322070B,
.22305230162323123406B,23717243472510525471B,25727256562570325614B,
.26002260202600125701B,25536253642511524676B,24362241052356223272B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=3,3),K=2,2) /
.24170235352257221605B,20576173521630115162B,14173133521260411777B,
.11767117661175611460B,11373114131130611263B,11267111501117711075B,
.10023101131016510321B,10660113061204312560B,13306143211537216447B,
.17357200342035120163B,20104201062032320422B,20702212672157722175B,
.22320227042332423770B,24401247122517125467B,25666260252604025751B,
.26156260632606725600B,25513252042503224636B,24351241402366023233B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=4,4),K=2,2) /
.24204234732271321771B,20445174021624215236B,14275134361256112172B,
.12055121571173211533B,11411113221122011221B,11324113161116711155B,
.10246102231036210520B,10754113301200512615B,13542146071537416372B,
.17361177352015720237B,20373202552035020540B,21006213502166522340B,
.22352226172335524274B,24725252502542225604B,25736261162617026210B,
.26212261432572425644B,25476252062502224662B,24431242262372523245B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=5,5),K=2,2) /
.24033232002241422045B,21136202241713116125B,15171143241364213165B,
.12624122141175411541B,11430114001133511310B,11511114011151211677B,
.10545106631107611400B,11764123161273413466B,14215147671566516461B,
.17170174202001220334B,20566206642074221064B,21166214132174622324B,
.22673226122351624216B,24654252242547425657B,25770257562602126101B,
.26042257532563525442B,25174250102455524463B,24272240402345623042B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=6,6),K=2,2) /
.23622232472271122355B,21503207372013517255B,16562155751476414341B,
.13742134241317612773B,12725125521264712707B,12571125131254512676B,
.11605120101227112537B,13050134631404014532B,15160156051615216527B,
.17221175601774320227B,20377205372067021102B,21254214362163422221B,
.22002224102303323440B,25734241532436424551B,24735251032500325025B,
.24747246072456524460B,24335242202377123611B,23455231652274122427B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=1,1),K=3,3) /
.24307234352252521534B,20357171111564214514B,13355124161147411015B,
.10614103721024510174B,10112102111010110101B,10243103521035010423B,
.07322074150745007571B,10141104021103211544B,12367133411456115653B,
.16511173161773320272B,20214203352061321006B,21301216052207422347B,
.23004234352361623563B,24020246342524325723B,26240265502672427132B,
.27172271662716226771B,26475261032560625321B,24765243622377123363B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=2,2),K=3,3) /
.24245234202254021413B,20347171771572014566B,13444124211155411111B,
.10626104661030310152B,10076101061007110142B,10215102411034110420B,
.07251073250745207604B,10110104401110511661B,12400134731450115570B,
.16541175512011620341B,20371205402072521144B,21356215772205022265B,
.22755234172371724031B,24372250332552526236B,26577267362700627114B,
.27235271672705226704B,26417261342555725241B,24775244172404123426B
./
```

```
 DATA (((ICZOFF (I,J,K),I=1,18),J=3,3),K=3,3) /
.23763232112225521256B,20204167701557614333B,13417123371150211117B,
.10665105731036610045B,07706077061004210140B,10244102311034510454B,
.07337073520756507620B,10124104561110411645B,12440134141450715440B,
.16475173311773520246B,20461206402074121130B,21356216422200422260B,
.22712232762403624367B,25021254752606726422B,26670270622715427170B,
.27174271422677426553B,26352261372557225251B,24747244362402323331B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=4,4),K=3,3) /
.23715230602212621077B,20006166661552214430B,13412123751160611145B,
.10760107161034710057B,07674075740765610047B,10077101631026710471B,
.07526074550764207733B,10232106031120111765B,12602134541451515365B,
.16325172331764720217B,20557207252103621205B,21361216142206222315B,
.22705233022405424700B,25461260502641126563B,26727270772722027240B,
.27214271232676526574B,26320261372556525310B,24772244422376023221B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=5,5),K=3,3) /
.23277224712164421016B,20226173111616315165B,14226133541256312140B,
.11465110221050510252B,10221101741013410254B,10442105611070611113B,
.10215102651044210725B,11203115141213612652B,13321141441504515664B,
.16401171451771520402B,20777212662142421570B,21710221012240022502B,
.23245240702120324575B,25233257262621326513B,26667267702700626721B,
.26650265542637526126B,25645253332503024736B,24445241212351323001B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=6,6),K=3,3) /
.23067224412176321227B,20560200321714016332B,15512147601416113453B,
.13074125451231412125B,12001117501175711735B,11730117341201412141B,
.11416115041163612071B,12363127641345314023B,14520152121565216244B,
.16675172751776520320B,20630207772115221304B,21446215642177022276B,
.22163227042333723726B,24250245022473325143B,25327254352546425462B,
.25332252142507324702B,24467242512410623733B,23550231762270622375B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=1,1),K=4,4) /
.23607230742222321174B,17755165461540414414B,13407124011166311457B,
.11121105551016707767B,07504073260724407100B,07133073240732007501B,
.06757070570715107355B,07645102661077011715B,12643136431464715522B,
.16223167261745320133B,20562210542134121575B,22047224002262223013B,
.23226235422371623746B,24141246652530525567B,26155266112720127410B,
.27510276222751427310B,27023265122613325575B,25226246632427723637B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=2,2),K=4,4) /
.23624230342222521126B,17731165551537614426B,13377124031174211522B,
.11242107141036310035B,07544073110721107117B,07207073020736307576B,
.06773070730721407421B,07711103301106011661B,12571136331473615451B,
.16237170021745120133B,20616210632134121540B,22005222262243722606B,
.23134235452402424146B,24432250212547126151B,26625271452743227473B,
.27552276362753727335B,26772264332607325576B,25310250302436423665B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=3,3),K=4,4) /
.23601230202211321074B,20002166411546214334B,13321123721174411563B,
.11440111301062610150B,07503072170707107070B,07170072770745707725B,
.07054071210723007422B,10003104021112211656B,12633137351465415420B,
.16241170551770120277B,20667211372140721631B,22020222422245322671B,
.23067235012410024437B,24771254552604426414B,26776272632750127532B,
.27600276172754627357B,27106265422611125617B,25335250302433623652B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=4,4),K=4,4) /
.23601227252200220650B,17626165231547314454B,13423125161205611675B,
```

```
.11675113111070710257B,07561072220704707100B,07233073240756107765B,
.07217072410733107531B,10121105051113212024B,13021136761450115407B,
.16347171101760020346B,20725212152144121642B,22006222372263522773B,
.23021234642417324632B,25303256272640426644B,27202274132753527552B,
.27562276002744627306B,27061265372613425666B,25331247052421423525B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=5,5),K=4,4) /
.23361225772172521055B,20162172231627315251B,14347135221301512410B,
.11736112710507101406B,07713075760754707541B,07663101051030210423B,
.07612100151011210441B,10677112701170712434B,13241141171474715547B,
.16301170761766520347B,20750213422162722030B,22242224032254523005B,
.22770234642422624706B,25371260442643427011B,27212273152740127374B,
.27362272302705626577B,26316260052546425207B,24707243642372723220B
./
 DATA (((ICZOFF (I,J,K),I=1,18),J=6,6),K=4,4) /
.23055224402173221172B,20417176721707516256B,15474147321417613524B,
.13041124541213611673B,11465113501130211314B,11344115071156411712B,
.11127112631145711607B,12113125271310613527B,14207147041541716036B,
.16611172101762120234B,20640212542151321646B,22052221542231022550B,
.22627232632363324170B,24515250612540525625B,25761261132615326146B,
.26155260042563025441B,25151247052450624273B,24015235502321622555B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=1,1),K=1,1) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=2,2),K=1,1) /
.20026200222005220057B,20066200732014120145B,20134200742016120141B,
.20153201672026620254B,20171201372016620127B,20051200552003617746B,
.17646177471761417506B,17476174751750317502B,17536175401754017512B,
.17604175111750217420B,17432175212003020175B,20217202632022120321B,
.20627203302012120042B,20037201652022120242B,20244202442022720174B,
.20120200412001720053B,20136201512011720004B,17746176641773017742B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=3,3),K=1,1) /
.17764200152011020117B,20156201542015520163B,20207202032030320366B,
.20352202652024320252B,20206202032016320104B,20013177351771717675B,
.20074200102002417745B,17634176051763017632B,17645176411753017447B,
.17401174601753117643B,17737200542027620642B,20656207012070720670B,
.20773204622037320327B,20355204062043020460B,20465204522037720321B,
.20237201232010520156B,20160202142014020074B,20047176441767217735B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=4,4),K=1,1) /
.20052200552006020072B,20061200272002620057B,20140202752040120372B,
.20325203242033320275B,20237201632010120056B,20001177071771417645B,
.20110200371777617670B,17631175741762017606B,17623176051753217625B,
.17556176011761117765B,20026202422044620651B,21101212022110321045B,
.21031207552061320601B,20612206422071420724B,20714206672063120546B,
.20457204152024320210B,20153203172027020204B,20152200512003620042B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=5,5),K=1,1) /
.20066200312006320115B,20145201512016620204B,20203202402032320377B,
.20474204232037720364B,20270201612006320005B,17754200041771117736B,
.20227201622021220220B,20201201512017120170B,20175201622015520250B,
```

```
.20353204232046420505B,20643210572121521347B,21553216632201222060B,
.21636217432172521724B,21733217322175521744B,21741217302170121647B,
.21614215352152721375B,21260211522103120716B,20553204442032220252B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=6,6),K=1,1) /
.21061210522103320677B,20601205312046720420B,20434204212047720515B,
.20557206052057520545B,20500204012030120224B,20144200772003017751B,
.20432204512044420436B,20432204402043120452B,20474205102052520621B,
.20675207342104321175B,21333215532175422121B,22352224242257122735B,
.22471226142250722566B,22563225612254422526B,22572226752271322715B,
.22620225752250622375B,22307221762202621706B,21513214022131621224B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=1,1),K=2,2) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=2,2),K=2,2) /
.20035200042003017774B,20003200522005720076B,20140201302012120101B,
.20103201032010320103B,20024200021773617743B,17766200302001620031B,
.20031177661776617626B,17632176171760017602B,17574175511752217511B,
.17510175021753517554B,17660177472012520304B,20437204762040320342B,
.20345202002000317716B,17723177512006520074B,20120201212013120044B,
.17772177401777520006B,20000177461773617656B,17640176531771317676B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=3,3),K=2,2) /
.20053200632012020060B,20044200642006120127B,20147201702024720307B,
.20245201622007420072B,20055200142001120004B,20004200322001617777B,
.20033200442003117757B,17706176261761017607B,17571175561754417445B,
.17552175721763217765B,20131202062034320667B,20726207132060720672B,
.20672204642033420213B,20177202332032620363B,20415203712030020220B,
.20140201332011020073B,20125201022006220037B,20017177751773717746B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=4,4),K=2,2) /
.20163202022023620221B,20563201452011320140B,20132202302035120400B,
.20373203062022520132B,20073201012004720054B,20033200222002420043B,
.20134201262007020027B,17764177521773417725B,17701176531772517711B,
.17720177512004120216B,20250204412066621005B,21027210512104521136B,
.21247210752074620616B,20547205652066120676B,20707206632057020507B,
.20432203232025220224B,20230202602024220243B,20222201522012120113B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=5,5),K=2,2) /
.20612211302110721031B,20351204232044020435B,20424204222047020533B,
.20577206112055720457B,20325202162054020443B,20441205212045420066B,
.20373204602046620465B,20475205112052220527B,20551206152057720572B,
.20612207112076421063B,21224213422152121637B,21744220522201222120B,
.22140223712206422066B,22143222152216322107B,22111220562202622000B,
.21662216152155221447B,21345212372113021002B,20724206762070520604B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=6,6),K=2,2) /
.21464214132137521320B,21234212362111721101B,21044210242105321077B,
.21115211132107721030B,20755206632057120515B,20450204232036320331B,
.20756207442076320772B,21006210122103021055B,21107211422120421255B,
.21351213712145421564B,21731220442217222320B,22421225442265622777B,
.22756230532306423042B,23005227652274722753B,22776227562272722730B,
```

```
.22734227032263622547B,22457223442224122120B,22001217012162121550B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=1,1),K=3,3) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=2,2),K=3,3) /
.20053200162002520032B,20052200672011720143B,20147201712021220170B,
.20132201012012120133B,20073201552011420026B,20062200762012020061B,
.20010200231777417765B,17740177451774117701B,17724177211767217730B,
.17725177231775420032B,20132202172030420476B,20534205012042620375B,
.20373202332010420051B,17774200152011720202B,20225202052016420151B,
.20126200772002017775B,20026200162001217766B,17756177601776517762B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=3,3),K=3,3) /
.20416201662016420150B,20212202672027420263B,20311203442035320331B,
.20326202632024320231B,20213201552007620113B,20127200652006720112B,
.20160201332013420137B,20110200732011020074B,20060200542004420047B,
.20076201272021520314B,20413205062067521050B,21125211032074620645B,
.20756206232036020234B,20176202262031720410B,20463204652043420431B,
.20361203032017020131B,20141201402007520102B,20061200552004720052B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=4,4),K=3,3) /
.20556206042033620334B,20345203652037120325B,20373204472044420444B,
.20443204102041720320B,20234202262011620077B,20073200572006620115B,
.20267202622025520304B,20244202362026420252B,20247202262022020271B,
.20331203732044120544B,20702210622126621402B,21374213142121721137B,
.21354211462077020654B,20577205742063720711B,20761210142077020733B,
.20650205352043720322B,20274202502024620234B,20225202052022020213B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=5,5),K=3,3) /
.21111210752110021077B,21060210412077320732B,20720207422072620717B,
.20732207272063520524B,20372202702022620142B,20064200672010520167B,
.20536205742063520663B,20712207522077020777B,21013210222101521034B,
.21144212132132021444B,21473217162200622070B,22151220502213522476B,
.22372221772161022121B,22170222012217222222B,22213221722214622102B,
.22043217442165121547B,21440213552126121150B,21066210332076720747B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=6,6),K=3,3) /
.22071220022172521666B,21557215052142421371B,21327212472127621316B,
.21304212362115221044B,20733206242052220501B,20467204612046120445B,
.21130211362113721154B,21176212262127321337B,21367214072144721541B,
.21626217102173721773B,22013221102223022350B,22433225142256522676B,
.23202231762316623157B,23130231262312723124B,23122230732305023063B,
.23056227742270022640B,22577225122240722310B,22150220542176621735B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=1,1),K=4,4) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
```

```
 DATA (((ICLOFF(I,J,K),I=1,18),J=2,2),K=4,4) /
.20075200632006520070B,20116201162012120161B,20154202052020520153B,
.20066200432006220147B,20124200622003120017B,20037200702004420006B,
.20047200662007520060B,20044200342003220025B,20030200462006120070B,
.20131201562017220212B,20233202732034720415B,20460204612041420367B,
.20271202032011220051B,17773200072004420111B,20135201342015620203B,
.20177201562011120067B,20035200412004420032B,20044200332002020030B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=3,3),K=4,4) /
.20377203662035520360B,20246202542030020273B,20320203742040320323B,
.20205203062013320221B,20234201662010720065B,20117200572005320070B,
.20175202042021220202B,20203201752017020165B,20200202102023020251B,
.20335203662042220464B,20535206242072521000B,21043210532103320736B,
.20637205152036520300B,20213202162026220330B,20411204512050320521B,
.20465204052031720253B,20213201602016320176B,20170201572014220145B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=4,4),K=4,4) /
.20533205272052420526B,20357204072041420374B,20467205362053320452B,
.20334203262033520325B,20275202522020220156B,20143201202014020133B,
.20330203212034420352B,20332203342034420336B,20331203432036720465B,
.20522205772063120705B,21041211622126321350B,21440214262134421231B,
.21201210562072220613B,20544205622056120617B,20664207322102021030B,
.20771206342056220506B,20447203612035520326B,20325203062030720301B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=5,5),K=4,4) /
.21222212302123121247B,21260212452122321173B,21160211242107521046B,
.20776207512066620614B,20522204632043220415B,20333202572022520177B,
.20573206402071220756B,21022210532107221077B,21123211652121421247B,
.21315214162150621614B,21706217762206622153B,22245223272225722247B,
.22407223142215022112B,22110221172206522066B,22121221172216122155B,
.22160221132206121766B,21656215642146221376B,21307212272120721166B
./
 DATA (((ICLOFF(I,J,K),I=1,18),J=6,6),K=4,4) /
.22163221462213322105B,22032217462167321644B,21611215262146221467B,
.21422213472127421157B,21100207632067720620B,20553205232047520475B,
.21122211532122621303B,21357214162147121543B,21606216612173522015B,
.22106221672226322346B,22430224272247622570B,22642227132275222777B,
.23253232252323423214B,23222232362326723307B,23266232742325523237B,
.23222232062314523054B,22766227172262622510B,22401222662215622075B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=1,1),K=1,1) /
.20427204462047320525B,20512204762053720553B,20664207272071321062B,
.21163212302127521304B,21340214042136521247B,21212211732107721002B,
.20356202002005617662B,17441171571675616567B,16421162771607616000B,
.15555154441532115151B,15063150041502015121B,15236151261526415562B,
.16027161471653317112B,17407177122027320606B,21064214022164621762B,
.21773220252175221711B,21667216002145321273B,21175207622064620465B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=2,2),K=1,1) /
.20347204122046620501B,20511205032051420650B,20762210412106721104B,
.21166212442124021254B,21353214212134721247B,21207211632112020756B,
.20421202322002617635B,17407172001704216666B,16460163261616716046B,
.15704155511534015170B,15073150241477015045B,15162151271530315561B,
.15616163051657317057B,17361177642024620550B,21051213722162621731B,
.22041221322202221746B,21731216132145321377B,21271210732071620636B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=3,3),K=1,1) /
.20421204322046520543B,20551206022065021020B,21130212142132321360B,
```

```
.21320213072124121274B,21331213502127721244B,21161211132102720736B,
.20252201031770317511B,17355171031675016560B,16404162651611615774B,
.15712154551523015064B,14763146001464314665B,14713150771527515652B,
.15740162651652416752B,17252176172017120506B,21000212652152521675B,
.22011221512205021762B,21712216312154121414B,21314211502103520714B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=4,4),K=1,1) /
.20424204142050320567B,20647207272101421065B,21170213162140321507B,
.21551213772132621342B,21335213212132021253B,21214210652077020671B,
.20250200461770017470B,17270170731673216535B,16403162561610716002B,
.15642154361517615060B,14715146301453414565B,14705151171530615703B,
.16053163171646116674B,17173175462012620504B,20767212242147221636B,
.21723217162174321667B,21610215602155321 13B,21334211752110520744B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=5,5),K=1,1) /
.20466204372046420522B,20555206432101221151B,21346214372152021501B,
.21514214562142621321B,21317212622121521110B,21036207602054120444B,
.20021175741746417257B,17066166521651016325B,16110160161572015573B,
.15321151501474714570B,14512145331440614272B,14353145201475215224B,
.15554160461642116702B,17164174371777620242B,20464206412073521011B,
.21162212362122221310B,21367213562126621066B,20777207542070520731B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=6,6),K=1,1) /
.20260202222010420266B,20401205162061220750B,21076211702124021272B,
.21347213472126721257B,21310212022111421017B,20673205702041120255B,
.20143177421746217266B,17056167011654616361B,16162160311565215533B,
.15361152031507115034B,15031150061503415063B,15133153151551315731B,
.16300165301700317230B,17351175171770720035B,20157202542035320450B,
.20534205322062120666B,20720206762067520675B,20715206242057620547B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=1,1),K=2,2) /
.20653206252070721007B,21060211642125721313B,21266213042126321240B,
.21254213152132521253B,21334214422146521433B,21373213632125621102B,
.20523203012010717644B,17410171151673416552B,16377162041577415542B,
.15303151021475414642B,14630146031457614610B,14667150561514115506B,
.15575161021647417141B,17470176722024620632B,21134214522175522212B,
.22205221352207722013B,22012217442164221500B,21443213312123620734B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=2,2),K=2,2) /
.20670206672077621025B,21073211412123121312B,21466213662141221362B,
.21341213172131121312B,21400214722146121455B,21440214162131721154B,
.20426202111776717524B,17313170611667116474B,16303161441600415533B,
.15521151001470714551B,14514145241450414554B,14700150651524615521B,
.15535161231646017026B,17267176172017420507B,21114214642200322104B,
.22133220712206222026B,22012217172164621561B,21460213532121121107B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=3,3),K=2,2) /
.20665206532074421030B,21105212012130521451B,21530215522163521610B,
.21453213342127621344B,21370214562150521451B,21434213362123121064B,
.20312201251766717421B,17172167541660616540B,16243161021570515515B,
.15263151011467614515B,14426144371447314417B,14623150341527115506B,
.15540161161640516567B,17051174252005220454B,21022213702167022010B,
.22054220252204221776B,21773217252163721552B,21430213112115521145B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=4,4),K=2,2) /
.21004207302076221044B,21173213052141021433B,21547216072166621733B,
.21635214402132421336B,21360214222147221427B,21355212702115421034B,
.20271200411762517403B,17141167311654116317B,16173160261560115455B,
```

```
.15311150701466714445B,14412143431437414437B,14542147431520615503B,
.15567161101653016454B,16745173101772420321B,20656211652147421715B,
.21775217422176622004B,21751216752163021535B,21423212772120121155B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=5,5),K=2,2) /
.21044210522111421111B,21105211272125121370B,21523216302167721701B,
.21624216042147521435B,21417213452127621166B,21101207212057620440B,
.20155177301745317213B,16755165331634716141B,15742156031547515376B,
.15224150171464314517B,14435144421446414513B,14543146461506115313B,
.15562157731630716561B,16772172761761020102B,20361205412071421113B,
.21167212542132721411B,21442214422133021247B,21247212402121121154B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=6,6),K=2,2) /
.20540205742060220670B,20720207352101621134B,21224213572146121477B,
.21543214762143321355B,21336213202121221100B,20726205662040020205B,
.20255200361762217361B,17130166731651116332B,16133157431554215307B,
.15302152331520115130B,15055150361507115165B,15307154471562715764B,
.16124163601661217070B,17274174561760617675B,20037202302036420440B,
.20532206022067120721B,20740207362076520756B,20736210272073520741B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=1,1),K=3,3) /
.21222213252144021563B,21606217102174521747B,21742217712175721670B,
.21571215532156221524B,21607215302153321515B,21444213732126221114B,
.20344200701762517337B,17060166141641416220B,16030156641551615354B,
.15005146141440514220B,14140141431417614272B,14417146521513415416B,
.15550160461642017042B,17325175112001720360B,20723212722163122064B,
.22167222212213722117B,22053220132176221660B,21545214472135621227B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=2,2),K=3,3) /
.21252213172144021536B,21600216522171521746B,22015220232204422023B,
.21725216112153621536B,21611216172156421470B,21446214132132021130B,
.20314200401755017254B,17017165521632616136B,16023156631543315266B,
.15025146131444614266B,14203141671423214310B,14451146551513715460B,
.15527160521640716754B,17147174101771420247B,20653213102166022024B,
.22123221712217722145B,22113220452177021703B,21571214472133121225B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=3,3),K=3,3) /
.21231212452136321505B,21561216352172722022B,22037220772214322132B,
.22043217212152521513B,21543215542152721507B,21421213202121521027B,
.20217177131744617174B,16727165051631116106B,15741155561534115151B,
.14772146351440614247B,14173141601422514324B,14451146641520415477B,
.15473160251634416552B,16724172141756620177B,20524211322151421744B,
.22114221622223322207B,22070220202176121664B,21556214172131421237B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=4,4),K=3,3) /
.21212212672135621460B,21620217042200522003B,21771221002213422160B,
.22055216722154521517B,21525214772142021420B,21305212112107220727B,
.20153176571741017147B,16652164531621516014B,15630154461527215100B,
.14726146031437114227B,14110141121412514265B,14503147331515615431B,
.15462160221624616357B,16562170631746320063B,20454210012134221631B,
.21744220502217622156B,22103220462173221671B,21550214232131021271B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=5,5),K=3,3) /
.21207213012141021422B,21376214532153521622B,21731220132205522011B,
.21735216612155621513B,21433213412121721110B,21004206752055320400B,
.20036174721723016776B,16527162741611315702B,15453153331517415072B,
.14750145531435514276B,14307144221447214512B,14670151631537115500B,
.15752161701432216373B,16624171061737017702B,20202204452072021160B,
```

```
.21355215102156521612B,21552215102143421367B,21334213312125721307B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=6,6),K=3,3) /
.20631207212075721071B,21113211412123621333B,21370214512152121533B,
.21542214722141421324B,21246211672111220763B,20640204672030120105B,
.20065176211737417156B,16730165271631516130B,15746155741542215244B,
.15061147701476115000B,15032150611511415171B,15305154601563716015B,
.16026162501647316724B,17111172671745117614B,17771201242025420402B,
.20567206472070620752B,20766210232103721101B,21127210712105721063B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=1,1),K=4,4) /
.21454215562170521773B,22005217752176421714B,21651216542162621560B,
.21550215532161521563B,21630216042156021475B,21371213232112420757B,
.20361200701755617246B,16760164751627316062B,15656154361514114717B,
.14535143321415414044B,13755137441401414131B,14305145461503615374B,
.15555161061645317056B,17300174611767220150B,20422206722113621465B,
.22002221422222222260B,22241221622213322055B,21751216422150421346B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=2,2),K=4,4) /
.21472215742167121757B,21743216612175421721B,21674216772170321642B,
.21605215672155021525B,21612216212157221511B,21400212602113220774B,
.20352200471750617213B,16734164401622116030B,15652154031513114742B,
.14536143351417114056B,13775137611402014132B,14271145351503115357B,
.15556161001637616745B,17125173371757220004B,20337205762117321505B,
.21733221122217522242B,22232221722212522036B,21735216232147021343B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=3,3),K=4,4) /
.21472215332164621745B,21730217272177521776B,21736217472176621723B,
.21631215772153721501B,21553216122153521433B,21332212012105120734B,
.20233177321742717124B,16671164151617715767B,15630153411507114713B,
.14510143431421514076B,13763137421402014130B,14323145751503015337B,
.15551160541635216575B,16744171361743717740B,20232205232110021442B,
.21676220672222022246B,22211221342207422012B,21721215662147621317B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=4,4),K=4,4) /
.21461215412166422012B,22055220622202621757B,21743217742201221763B,
.21630215342150321475B,21507215252151021401B,21244211162076520635B,
.20161177051737117120B,16626163561615715723B,15520152771507314656B,
.14477143401422614105B,14007137511400314115B,14301145271477015303B,
.15526160211625516431B,16607170501733717623B,20175204142101521304B,
.21542220702214222225B,22213221472204621754B,21664215732146521353B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=5,5),K=4,4) /
.21430215012153421556B,21572216032163521705B,21761220032200421737B,
.21607216422156621545B,21473213772120621146B,21002206702056520456B,
.20023175061721516777B,16466162361604615637B,15441152421506114722B,
.14602144331432114232B,14167141621421414305B,14420146131504315316B,
.15534157721613716325B,16547167271715517405B,17700201112045020660B,
.21135213062147721573B,21604215702146521456B,21407213672140021412B
./
 DATA (((ICMOFF(I,J,K),I=1,18),J=6,6),K=4,4) /
.21031211222124221326B,21346213502140521462B,21514215202152021511B,
.21515215122145621405B,21341212462115321032B,20721205522042120240B,
.20107176731742417211B,16734164671625216045B,15670155111533415204B,
.15042147571467614643B,14646147171500515072B,15174153461555716032B,
.16017162371642016606B,16767171371730117450B,17602177762012720272B,
.20411206112070020750B,21026211052113721145B,21174211742115021200B
./
```

```
 DATA (((ICNOFF(I,J,K),I=1,18),J=1,1),K=1,1) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=2,2),K=1,1) /
.20152201422012120136B,20123201052007020104B,20105201022002020002B,
.17751177541770117666B,17665176211757717513B,20030200211761517613B,
.20024200322002120027B,20051200322003120036B,20052200612010320074B,
.20104201012010720115B,20156201462012020104B,20022200502007020000B,
.17700177722000720047B,20066200602003517757B,17725176741767617721B,
.17736177141772520003B,20017200432006720042B,20063200262005320107B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=3,3),K=1,1) /
.20305203422031420313B,20326203232034120311B,20227202422022120142B,
.20125200762002617735B,17732177131763717617B,17734177001767717715B,
.20071201322010420067B,20123201122015020150B,20161201502014420151B,
.20202201642016320150B,20162201262005320101B,20052200652006720074B,
.17737200562012320170B,20211201262006020016B,20002177601773017723B,
.17722177411777720037B,20065201022011720143B,20172201322023020272B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=4,4),K=1,1) /
.20512205352055520562B,20546205342054720524B,20424203342032220317B,
.20264202242014120102B,20042200321775217717B,17776200171777520015B,
.20143201372012720117B,20140201432020520172B,20170201562016620227B,
.20226202322020520173B,20162201312007120105B,20126201232013520143B,
.20034201162024120257B,20224201762016120145B,20115200712003620014B,
.20012200102006620125B,20173202232024420266B,20304203152037720456B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=5,5),K=1,1) /
.21270213402137121355B,21323212722117421115B,21132211612113521051B,
.20740206342056220513B,20442204342047420524B,20514204272044420455B,
.20435204462040620343B,20334204032036220361B,20356203502033220277B,
.20240202072016720151B,20146201472013520143B,20145201532022120265B,
.20251202762026220301B,20344203552040020402B,20402204112041220422B,
.20435205222054320572B,20627207022074221031B,21036210662114421241B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=6,6),K=1,1) /
.21644217162175021675B,21643216132156021471B,21420214222145121405B,
.21321212342116621132B,21072210422102721065B,21063211052107321064B,
.20710207042070220653B,20613205412050720467B,20443204132036620360B,
.20337203342033020331B,20335203322030520312B,20300202472027120360B,
.20277203442037420456B,20523205522061720641B,20675207512107521151B,
.21201212402125521255B,21302213172135021426B,21462215312161621626B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=1,1),K=2,2) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=2,2),K=2,2) /
.20200201712017420202B,20177201612013220111B,20116200702005720051B,
```

```
.20025200252000717724B,17742177131767417655B,17675201021764317703B,
.20016200202005420112B,20105201232013320135B,20135201332012020137B,
.20131201232011020045B,20056200442003020017B,20007177752000117761B,
.17715200012002420061B,20061200552000117745B,17727177171774320001B,
.17766177211773220021B,20042200752011020065B,20110201132010720113B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=3,3),K=2,2) /
.20402204332043220407B,20422204032040320340B,20306202342024220171B,
.20157201402010320032B,20021200102000317745B,20000177431776120007B,
.20132201512014320147B,20167202122023420227B,20222201752017020217B,
.20172201622012120102B,20071200712005720011B,20004200032000317760B,
.17744201122015420202B,20170201232003120001B,17770177521772717734B,
.17727177311777720077B,20140201742020220216B,20226202512026320306B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=4,4),K=2,2) /
.20602206372064720645B,20615206172053420555B,20504204242041220370B,
.20321202762024420175B,20150201052007720037B,20051201152012420131B,
.20267202402025420265B,20304203202033220320B,20311202502026720243B,
.20247202412017620130B,20105200672004720051B,20057200662006620052B,
.20052202142027720260B,20224201662014220131B,20103200602002620002B,
.20007200262010520215B,20254203062034420373B,20403204262047220502B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=5,5),K=2,2) /
.21362213562134321363B,21405213602130421234B,21232212412121021145B,
.21062210112075320703B,20633206012057720620B,20647206322063320620B,
.20642205602056220546B,20536205472056720550B,20507204602043620402B,
.20341202612022520214B,20173201762017020164B,20162201762024120306B,
.20356204542040620420B,20445204702047420510B,20516205152052320543B,
.20543205742064120664B,20720207702102721056B,21104211472121421251B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=6,6),K=2,2) /
.21747217622175421715B,21741217212170121643B,21566215712157721515B,
.21447214012134321303B,21275213222131121274B,21276213012127121255B,
.21153211502112021045B,21004207232066320623B,20563205462051420475B,
.20425204012037020372B,20413204202040520366B,20360203552037320462B,
.20460205102056320640B,20706207302077421025B,21050211022113721233B,
.21274212732133121364B,21377214162145321470B,21534216022166221645B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=1,1),K=3,3) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=2,2),K=3,3) /
.20216202152021320200B,20174201552014020123B,20153201502015620200B,
.20171201652014420120B,20153200622002720045B,20024200032001617744B,
.20060200762012120141B,20150201562016120155B,20153201522013120134B,
.20127201272013120113B,20052200342003020003B,17762177571774617727B,
.17746200212006120112B,20133201472013620107B,20022200042000320007B,
.20014200202005220115B,20135201452014020144B,20143201422014020143B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=3,3),K=3,3) /
.20426204442044020415B,20400203552035320335B,20353203322033120331B,
.20334203342027420242B,20240201662011420065B,20131201272010420050B,
.20146201552022120243B,20254202702030420303B,20302202662025320240B,
```

```
.20231202402020620157B,20115200622002717776B,17760177541775717747B,
.17764200552021120234B,20260202602024220201B,20133200742005420047B,
.20073201132013720212B,20254202712027720306B,20312203112031620325B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=4,4),K=3,3) /
.20655206532066120643B,20615205752053320545B,20542205322052220517B,
.20501204642043520406B,20360202742023320204B,20216202162016120173B,
.20263202702031620357B,20401204132042120416B,20413204012036320331B,
.20321203132025520213B,20157201262006720026B,20022200242003720025B,
.20066202552034320367B,20366203662030220262B,20256202232017620175B,
.20224202522030720342B,20401204312044720472B,20475205042052220517B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=5,5),K=3,3) /
.21442214332140521411B,21423213552131621276B,21255212522123321165B,
.21134211122107421031B,21010207622075620754B,20754207342071720666B,
.20710206522065320656B,20674206772070220671B,20637206212057220532B,
.20457204232035620275B,20276202432023420235B,20214202252031320535B,
.20571205712025420571B,20603206172061320633B,20656206662070120727B,
.20736207642100621023B,21060211052113221143B,21162212032122521251B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=6,6),K=3,3) /
.22062217752175421706B,21734217322170221656B,21611215752156321503B,
.21465214362137621333B,21325213412132621273B,21267212712126021246B,
.21233212042115421126B,21006210202076520731B,20702206502061520574B,
.20551205222050120446B,20436204212041720406B,20371203712042120475B,
.20643206602067720732B,20756210042103521067B,21112211462122321266B,
.21332213502136721376B,21406214112143021447B,21455215452156021574B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=1,1),K=4,4) /
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B,
.17777177771777717777B,17777177771777717777B,17777177771777717777B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=2,2),K=4,4) /
.20234202302022720206B,20157201532016420150B,20154201542017420213B,
.20226202362022520163B,20164201442010220066B,20067200532004720052B,
.20076201052012120143B,20153201642015620151B,20154201432013120125B,
.20116201262014120132B,20121201072006220037B,20017177751777117772B,
.20025200412006520110B,20120201342014720135B,20121201052005520051B,
.20064200662013720155B,20157201672017420200B,20200201722016220175B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=3,3),K=4,4) /
.20450204452043020376B,20360203352031420334B,20342203312034120371B,
.20404203672036020314B,20317203212024720210B,20156201602013220062B,
.20167202162025520275B,20303203162032520314B,20273202572023620233B,
.20212202202023420225B,20174201412011520065B,20037200162001020014B,
.20077201552021220240B,20263202732027620270B,20243202002014020126B,
.20150201772026020312B,20335203462035520357B,20372203732037220375B
./
 DATA (((ICNOFF(I,J,K),I=1,18),J=4,4),K=4,4) /
.20677206652063320603B,20570205512053620523B,20517205222052220533B,
.20520204662047720453B,20465204532040420313B,20312203112027320233B,
.20272203152036420415B,20434204532045720450B,20416203752036220333B,
.20331203202031420300B,20241202032015520115B,20071200662007720104B,
.20176202622034520356B,20373203742040620377B,20354203312025520232B,
```

```
     .20242203242040220444B,20476205352055020560B,20566205712057720570B
     ./
      DATA (((ICNOFF(I,J,K),I=1,18),J=5,5),K=4,4) /
     .21473214562143521406B,21360213252127621274B,21243212242120021164B,
     .21173211562113221106B,21105210642104221042B,21036210262101520766B,
     .20722207302073720735B,20726207462075420746B,20726207112065620601B,
     .20530204602041420360B,20346203432034620342B,20335203362034520367B,
     .20547206112063520642B,20644206622067020707B,20717207352076520773B,
     .21020210412106721122B,21170212212125721275B,21320213322134021344B
     ./
      DATA (((ICNOFF(I,J,K),I=1,18),J=6,6),K=4,4) /
     .22065220572202021754B,21763217542173421714B,21660216242160021561B,
     .21542215222150121430B,21430214352141421404B,21350213412133321303B,
     .21232212162121021174B,21145211262107121036B,21013207652073020703B,
     .20641206132057620563B,20545205202050620501B,20505204762050220513B,
     .20700207342076721012B,21041210662111021135B,21161212342131021366B,
     .21414214712152021544B,21547215412154521557B,21574216102167321647B
     ./
C
C
C  ***************************
C  *****  INITIALIZATION  *****
C  ***************************
C
      IF(ICCAL.NE.1) GO TO 40
C
C  SET CERTAIN UNINITIALIZED PARAMETERS TO ZERO ......
C
      DO 5 I=1,3
      IF(XEM(I).EQ.0.99999) XEM(I) = 0
      IF(SRA(I).EQ.0.99999) SRA(I) = 0
      DO 5 J=1,3
    5 IF(DSA(J,I).EQ.0.99999) DSA(J,I) = 0
C
      IF(OFF.EQ.0.99999) OFF = 1.
      IF(UP.EQ.0.99999) UP = 1.
      IF(ZWS.EQ.0.99999) ZWS = 0
      IF(CDX.EQ.0.99999) CDX = 0
      IF(ECX.EQ.0.99999) ECX = 1.
      IF(ECY.EQ.0.99999) ECY = 1.
      IF(ECZ.EQ.0.99999) ECZ = 1.
      IF(CLP.EQ.0.99999) CLP = 0.
      IF(CMQ.EQ.0.99999) CMQ = 0.
      IF(CNR.EQ.0.99999) CNR = 0.
      IF(RON.EQ.0.99999) RON = 0
C
C  SET UP CONSTANTS FOR THE BOUNDRY LAYER PLANE EQUATION ......
C
      CONS(1) = CONS(2) = 0
      CONS(3) = 1.
C
C  SET UP THE CENTROID VECTOR ......
C
      CENT(1) = CDX
      CENT(2) = CENT(3) = 0
C
C  DETERMINE THE HYDRAULIC DIAMETER ......
C
```

```
      HD = SQRT(4.*S/3.14159)
C
C  CALCULATE THE CENTROID TABLE ......
C
      DO 10 I=1,20
 10   TCZ(I) = 0
      WRITE(6,15)
 15   FORMAT(//5X,*--- CENTROID TABLE CALCULATED FOR COMPONENT*,
     .       * AS ---*,//16X,*LENGTH*,8X,*CENTROID*/)
      NTAE = TAE(2)
      DO 20 I=2,NTAE
      K=NTAE+2+I
 20   TCZ(I)=(TCZ(I-1)*TAE(K)+.5*(TAE(K+1)-TAE(K))*
     .       (TAE(3+I)+TAE(2+I)))/TAE(K+1)
      WRITE(6,30) (TAE(I+3),TCZ(I),I=1,NTAE)
 30   FORMAT(16X,F5.2,10X,F7.4)
C
      ALPHA = BETA = VMACH = Q = EXL = EXA = 0
      CX = CY = CZ = CL = CM = CN = 0
C
C  ==================================================
C
C  ZERO OUT THE AERO FORCES AND TORQUES ......
C
 40   DO 50 I=1,3
 50   F(I) = T(I) = 0
C
C  BYPASS ROUTINE DURING STEADY STATE WITH THE RAIL COMPONENT IN THE
C  MODEL ......
C
      IF (INST.EQ.31 .AND. OFF.EQ.0) GO TO 110
C
      IF (OFF.EQ.1.) GO TO 60
C
C  CALCULATE XEM IN THE AIRPLANE SYSTEM ......
C
      CALL VECXYZ (XEMA,XEM,SRA,DSA,2)
C
C  CHECK TO SEE IF SEAT HAS PENETRATED THE BOUNDRY LAYER ......
C
      IF(ZWS*UP.LE.XEMA(3)*UP) GO TO 110
C
C  CONVERT FROM DEGREES TO RADIANS ......
C
 60   DO 70 I=1,3
 70   ESTIR(I) = EST(I) * RPD
C
C  DETERMINE ATMOSPHERIC PROPERTIES ......
C
      CALL ATMOS (VS,RHO,-SRP(3),UW,0,0,0)
C
C  PUT THE WIND INTO THE BODY COORDINATES ......
C
      CALL DIRCOS (DES,ESTIR)
      CALL MATMPY (UWB,DES,UW,3,3,1)
C
C  ADD THE WIND VELOCITY TO THE SEAT VELOCITY ......
C
```

```
      UO(1) = UST(1)-UWB(1)
      UO(2) = UST(2)-UWB(2)
      UO(3) = UST(3)-UWB(3)
C
C  DETERMINE THE AERO VEARIABLES ......
C
      IF(UO(1).EQ.0.0.AND.UO(3).EQ.0.0) UO(1)=.01
      ALPHA = ARTAN2(UO(3),UO(1))*DPR
      CALL DOTPRD (VBAR2,UO,UO,3)
      VBAR = SQRT(VBAR2)
      BETA = ASIN(UO(2)/VBAR)*DPR
      VMACH = VBAR/VS
      Q = .5 * RHO * VBAR2
C
C  PERFORM TABLE LOOKUP FOR AERODYNAMIC COEFFICIENTS ......
C
      TBLALPH = ALPHA
      IF(ALPHA .LT. 0.0) TBLALPH = ALPHA + 360.0
      TBLBETA = ABS(BETA)
      IF(RON.EQ.0.) CALL TLU (ICXOFF,72,6,4,ALF,BET,AMACH,TBLALPH,
     .                        TBLBETA,VMACH,COEF,6)
      IF(RON.NE.0.) CALL TLU (ICXON,72,6,4,ALF,BET,AMACH,TBLALPH,
     .                        TBLBETA,VMACH,COEF,6)
      CX = COEF(1)
      CY = -COEF(2) * SIGN(1.,BETA)
      CZ = COEF(3)
      CL = -COEF(4) * SIGN(1.,BETA)
      CM = COEF(5)
      CN = -COEF(6) * SIGN(1.,BETA)
C
C  BYPASS EMERGE CALCULATIONS IF SEAT IS OFF RAILS
C
      IF(OFF.EQ.1.) GO TO 90
C
C  ************************************************************
C  **  CALCULATE THE AERODYNAMIC FORCES AND TORQUES ACTING ON  **
C  **  THE SEAT/MAN AS IT IS EMERGING FROM THE AIRPLANE ...... **
C  ************************************************************
C
C  CALCULATE THE SEAT Z-AXIS UNIT VECTOR DIRECTION COSINES WITH
C  RESPECT TO THE AIRPLANE SYSTEM ......
C
      DO 80 I=1,3
 80   DC(I) = DSA(I,3)
C
C  CALCULATE THE POINT OF INTERSECTION BETWEEN THE BOUNDRY
C  LAYER PLANE AND THE LINE THAT BOTH PASSES THROUGH XEMA AND
C  IS PARALLEL WITH THE SEAT SYSTEM Z AXIS ......
C
      CONS(4) = -ZWS
      CALL LINEPL (XI,CONS,XEMA,DC)
C
C  DETERMINE THE SEAT/MAN EXPOSED LENGTH ......
C
      EXL=SQRT((XI(1)-XEMA(1))**2+(XI(2)-XEMA(2))**2+(XI(3)-XEMA(3))**2)
C
C  CALCULATE THE EXPOSED AREA FROM THE TABLE ......
C
```

```
      EXA = TBLU1(EXL,TAE(4),TAE(NTAE+4),1,-NTAE)
C
C  CALCULATE THE AERO FORCES FROM THE AERO COEFFICIENTS, THE
C  EXPOSED AREA, AND THE EMERGENCE COEFFICIENTS ......
C
      QAREA = Q * EXA
      F(1) = CX * QAREA * ECX
      F(2) = CY * QAREA * ECY
      F(3) = CZ * QAREA * ECZ
C
C  CALCULATE THE Z-AXIS POSITION OF THE CENTROID ......
C
      CENT(3)=XEM(3)-SIGN(1.,XEM(3))*TBLU1(EXL,TAE(4),TCZ(1),1,-NTAE)
C
C  CALCULATE THE RAIL/SEAT TORQUES ......
C
      CENT(1) = CDX
      CALL CRSPRD (T,CENT,F)
C
      GO TO 110
C
C  ///////////////////////////////////////////////////////////////////
C
C  ADD DAMPING TERMS FOR AN AIRSPEED GREATER THAN .1 FT/SEC
C
 90   IF(VBAR.LE.0.1) GO TO 100
C
      HDO2V = HD/(VBAR+VBAR)
C
C  ADD ROLL DAMPING ......
C
      CL = CL + CLP * WST(1) * HDO2V
C
C  ADD PITCH DAMPING ......
C
      CM = CM + CMQ * WST(2) * HDO2V
C
C  ADD YAW DAMPING ......
C
      CN = CN + CNR * WST(3) * HDO2V
C
C  COMPUTE THE AERO FORCES AND MOMENTS ABOUT THE SRP ......
C
 100  QS = Q * S
      F(1) = CX * QS
      F(2) = CY * QS
      F(3) = CZ * QS
      T(1) = CL * QS * HD
      T(2) = CM * QS * HD
      T(3) = CN * QS * HD
C
 110  RETURN
      END
```

```
      SUBROUTINE CS (AIL,AILDOT,IAIL,ELE,ELEDOT,IELE,RUD,RUDDOT,IRUD,
     .               COA,TCA,TDA,COE,TCE,TDE,COR,TCR,TDR,TRM)
C
C         ----- EASIEST AIRPLANE CONTROL SURFACE COMPONENT -----
C
C  DESIGNED BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C
C  ***************  CS OUTPUTS  ***************
C
C  AIL    - AILERON DEFLECTION FROM TRIM POSITION (DEG)
C  AILDOT - AILERON RATE (DEG/SEC)
C  IAIL   - INTEGRATION CONTROL
C
C  ELE    - ELEVATOR DEFLECTION FROM TRIM POSITION (DEG)
C  ELEDOT - ELEVATOR RATE (DEG/SEC)
C  IELE   - INTEGRATION CONTROL
C
C  RUD    - RUDDER DEFLECTION FROM TRIM POSITION (DEG)
C  RUDDOT - RUDDER RATE (DEG/SEC)
C  IRUD   - INTEGRATION CONTROL
C
C  ***************  CS INPUTS  ***************
C
C  COA - AILERON COMMANDED POSITION (DEG)
C  TCA - AILERON TIME CONSTANT (SEC)
C  TDA - TIME DELAY AFTER WHICH THE AILERON RATE IS CALCULATED (SEC)
C
C  COE - ELEVATOR COMMANDED POSITION (DEG)
C  TCE - ELEVATOR TIME CONSTANT (SEC)
C  TDE - TIME DELAY AFTER WHICH THE ELEVATOR RATE IS CALCULATED (SEC)
C
C  COR - RUDDER COMMANDED POSTION (DEG)
C  TCR - RUDDER TIME CONSTANT (SEC)
C  TDR - TIME DELAY AFTER WHICH THE RUDDER RATE IS CALCULATED (SEC)
C
C  TRM(4) - AIRPLANE THRUST AND CONTROL SURFACE POSITIONS AT TRIM
C                TRM(1) - ENGINE THRUST (LB)  -- NOT USED --
C                TRM(2) - AILERON POSITON (DEG)
C                TRM(3) - ELEVATOR POSITION (DEG)
C                TRM(4) - RUDDER POSITION (DEG)
C
C
      DIMENSION TRM (4)
      COMMON /CTIME/ TIME
      COMMON /CICCAL/ ICCAL
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
C  ****************************
C  *****  INITIALIZATION  *****
C  ****************************
C
      IF(ICCAL.NE.1) GO TO 10
C
      IF(COA.EQ.0.99999) COA = 0
      IF(COE.EQ.0.99999) COE = 0
      IF(COR.EQ.0.99999) COR = 0
```

```
C
      IF(TCA.EQ.0.99999) TCA = 0
      IF(TCE.EQ.0.99999) TCE = 0
      IF(TCR.EQ.0.99999) TCR = 0
C
      IF(TDA.EQ.0.99999) TDA = 0
      IF(TDE.EQ.0.99999) TDE = 0
      IF(TDR.EQ.0.99999) TDR = 0
C
C  /////////////////////////////////////
C
C  *****  AILERON  *****
C
 10   IF(TCA.LE.0) AILD = 0
      IF(TCA.GT.0) CALL LAG (AILD,COA,AIL,TRM(2),TCA,TIME,TDA)
      IF(IAIL.NE.0) AILDOT = AILD
C
C  *****  ELEVATOR  *****
C
      IF(TCE.LE.0) ELED = 0
      IF(TCE.GT.0) CALL LAG (ELED,COE,ELE,TRM(3),TCE,TIME,TDE)
      IF(IELE.NE.0) ELEDOT = ELED
C
C  *****  RUDDER  *****
C
      IF(TCR.LE.0) RUDD = 0
      IF(TCR.GT.0) CALL LAG (RUDD,COR,RUD,TRM(4),TCR,TIME,TDR)
      IF(IRUD.NE.0) RUDDOT = RUDD
C
      RETURN
      END
```

```
      SUBROUTINE CT (TCP,
     .               EF,EFDOT,IEF,EL,ELDOT,IEL,WK,WKDOT,IWK,
     .               WB,WBDOT,IWB,
     .               FL,FON,FCA,TCA,FCS,TCS,CF,CEX,CV,TLO,PC,R,CVH,TSO,
     .               FSO,SW,UP,SAP,AAP,UCL,CSK,VI,PA,PT,CBP,C,CI,PMW,SK,
     .               CK,GAM,TF,C1,C2,B,BXP,TI,TOE,SRP,UST,EST,WST,XAP,
     .               UAP,EAP,WAP)
C
C  **************  EASIEST CATAPULT COMPONENT  **********
C
C
C  DESIGNED BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C  FORCES AND MOMENTS ACTING ON THE VEHICLE AND THE SEAT FROM
C  A CLOSED TELESCOPING TUBE CATAPULT
C
C  ***************  CATAPULT TABLES  ***************
C
C     TCP - CATAPULT PROPELLANT CONSUMPTION TABLE
C
C             THE INDEPENDENT VARIABLE IS THE PROPELLANT
C             WEB CONSUMED (IN) AND THE DEPENDENT VARIABLE
C             IS THE PROPELLANT CONSUMED (SLUGS)
C
C  ***************  CATAPULT OUTPUTS  ***************
C
C  INTERNAL FRICTION ENERGY ......
C
C     EF    - INTERNAL FRICTION ENERGY (FT-LB)
C     EFDOT - INTERNAL FRICTION ENERGY RATE (FT-LB/SEC)
C     IEF   - INTEGRATION CONTROL
C
C  HEAT LOSS ......
C
C     EL    - HEAT LOSS (FT-LB)
C     ELDOT - HEAT LOSS RATE (FT-LB/SEC)
C     IEL   - INTEGRATION CONTROL
C
C  CATAPULT WORK ......
C
C     WK    - CATAPULT WORK (FT-LB)
C     WKDOT - CATAPULT WORK RATE (FT-LB/SEC)
C     IWK   - INTEGRATION CONTROL
C
C  PROPELLANT WEB BURNED ......
C
C     WB    - PROPELLANT WEB BURNED (IN)
C     WBDOT - PROPELLANT WEB BURN RATE (IN/SEC)
C     IWB   - INTEGERATION CONTROL
C
C     FL     - CATAPULT MODE FLAG
C                0 = PRIOR TO INITIATION
C                1 = CATAPULT IGNITION UP TO STRIPOFF
C                2 = CATAPULT STRIPOFF
C                3 = CATAPULT OFF
C     FON    - STRIPOFF FLAG FOR SUSTAINER ROCKET COMPONENT
C               (1 = ROCKET ON)
```

```
C     FCA(3) - X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS OF THE
C              CATAPULT ON THE AIRPLANE (LB)
C     TCA(3) - X,Y,Z AIRPLANE BODY AXIS TORQUE COMPONENTS OF THE
C              CATAPULT ON THE AIRPLANE (FT-LB)
C     FCS(3) - X,Y,Z SEAT BODY AXIS FORCE COMPONENTS OF THE
C              CATAPULT ON THE SEAT (LB)
C     TCS(3) - X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS OF THE
C              CATAPULT ON THE SEAT (FT-LB)
C     CF     - CATAPULT FORCE MAGNITUDE (LB)
C     CEX    - CATAPULT EXTENSION (FT)
C     CV     - CATAPULT EXTENSION VELOCITY (FT/SEC)
C     TLO    - INITIAL LENGTH OF CATAPULT PRESSURE CHAMBER (IN)
C     PC     - CIRCUMFERENCE OF CATAPULT PRESSURE CHAMBER (IN)
C     R      - GAS CONSTANT  (FT-LBF/SLUG-K)
C     CVH    - CONSTANT VOLUME SPECIFIC HEAT (FT-LBF/SLUG-K)
C     TSO    - CATAPULT STRIPOFF TIME (SEC)
C     FSO    - CATAPULT FORCE AT STRIPOFF (LB)
C
C  ***************  CATAPULT INPUTS  ***************
C
C     SW     - FLAG FOR CATAPULT IGNITION ( 1 = CATAPULT ON )
C     UP     - EJECTION DIRECTION FLAG WRT THE AIRPLANE
C                +1 = UPWARD EJECTION
C                -1 = DOWNWARD EJECTION
C     SAP(3) - SEAT ATTACHMENT POINT FOR THE CATAPULT (FT)
C     AAP(3) - AIRPLANE ATTACHMENT POINT FOR THE CATAPULT (FT)
C     UCL    - UNLOADED CATAPULT LENGTH (FT)
C     CSK    - CATAPULT STROKE (FT)
C     VI     - INITIAL FREE VOLUME (IN**3)
C     PA     - PISTON AREA (IN**2)
C     PT     - TANG RELEASE PRESSURE (LB/IN**2)
C     CBP    - CATAPULT BURST PRESSURE (LB/IN**2)
C     C      - MASS OF TOTAL PROPELLANT (SLUGS)
C     CI     - IGNITER PROPELLANT MASS (SLUGS)
C     PMW    - PROPELLANT MOLECULAR WEIGHT (LB/(LB-MOLE))
C     SK     - CATAPULT SPRING CONSTANT (LB/FT)
C     CK     - CATAPULT DAMPING CONSTANT (LB/FT/SEC)
C     GAM    - RATIO OF SPECIFIC HEATS
C     TF     - CONSTANT VOLUME FLAME TEMPERATURE (DEG K)
C     C1     - FRICTION PROPORTIONALITY CONSTANT
C     C2     - HEAT LOSS CONSTANT
C     B      - BURN RATE PROPORTIONALITY CONSTANT (IN/SEC/(LB/IN**2))
C     BXP    - BURN RATE EXPONENT
C     TI     - CATAPULT TEMPEATURE PRIOR TO IGNITION (DEG K)
C     TDE    - CATAPULT FORCE DECAY TIME (SEC)
C     SRP(3) - X,Y,Z EARTH SYSTEM POSTION VECTOR OF THE
C              SEAT REFERENCE POINT (FT)
C     UST(3) - X,Y,Z SEAT BODY AXIS VELOCITY VECTOR OF THE
C              SEAT (FT/SEC)
C     EST(3) - EARTH TO SEAT EULER ANGLES (DEG)
C     WST(3) - X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY
C              OF THE SEAT (DEG/SEC)
C     XAP(3) - X,Y,Z EARTH SYSTEM POSITION VECTOR OF THE
C              AIRPLANE (FT)
C     UAP(3) - X,Y,Z AIRPLANE BODY AXIS VELOCITY VECTOR OF
C              THE AIRPLANE (FT/SEC)
C     EAP(3) - EARTH TO AIRPLANE EULER ANGLES (DEG)
C     WAP(3) - X,Y,Z AIRPLANE BODY AXIS ANGULAR VELOCITY
```

```
C               OF THE AIRPLANE (DEG/SEC)
C
C   DIMENSIONS OF CALLING ARGUMENTS ......
C
      DIMENSION FCA(3),TCA(3),FCS(3),TCS(3),SAP(3),AAP(3),
     .          SRP(3),UST(3),EST(3),WST(3),XAP(3),UAP(3),
     .          EAP(3),WAP(3)
C
C   INTERNAL DIMENSIONS
C
      DIMENSION DES(3,3),DSE(3,3),DEA(3,3),DAE(3,3),
     .          SAPE(3),AAPE(3),DXL(3),EXT(3),USAPE(3),
     .          UAAPE(3),CDV(3),FCP(3),FSS(3),FSD(3),
     .          FC(3),CXUV(3),ESTIR(3),WSTIR(3),EAPIR(3),WAPIR(3)
C
      COMMON / CTIME /TIME
      COMMON / CICCAL / ICCAL
      COMMON / COVRLY / INST
      COMMON / CSSFLG / SSFLG
      COMMON / CIO / IREAD,IWRITE,IDIAG
C
      DATA RPD / .01745329 /
C
C   **************************
C   *****  INITIALIZATION  *****
C   **************************
C
      IF(ICCAL.NE.1) GO TO 10
C
C   COMPUTE THE INITIAL LENGTH (TLO) AND CIRCUMFERENCE (PC) OF THE
C   CATAPULT PRESSURE CHAMBER ......
C
      TLO = V1/PA
      PC = 2*SQRT(3.14159*PA)
C
C   CALCULATE THE GAS CONSTANT (R) AND THE CONSTANT VOLUME
C   SPECIFIC HEAT (CVH) ......
C
      R = 69475.694/PMW
      CVH = R/(GAM-1.0)
C
      TYPE = 8HCATAPULT
      CF = FL = TSO = FSO = FON = 0
      IF(UP.EQ.0.99999) UP = 1.0
      IF(TDE.EQ.0.99999) TDE = 0
C
      DO 5 I=1,3
 5    FCA(I) = TCA(I) = FCS(I) = TCS(I) = 0
C
C   ***************************************************
C
C   BYPASS THE REMAINING CODE IF THE CATAPULT IS PAST THE
C   STRIPOFF POINT ......
C
 10   IF(FL.EQ.3.) GO TO 170
      FCP(1) = FCP(2) = FCP(3) = 0
C
C   CHANGE ANGULAR STATES FROM DEGREES TO RADIANS ......
```

```
C
      DO 20 I=1,3
      ESTIR(I) = EST(I) * RPD
      WSTIR(I) = WST(I) * RPD
      EAPIR(I) = EAP(I) * RPD
 20   WAPIR(I) = WAP(I) * RPD
C
C  ****************************************************************
C  *                                                              *
C  *     DETERMINE THE VARIABLES CALCULATED FROM THE              *
C  *     EARTH POSITIONS OF THE AIRPLANE ATTACHEMENT              *
C  *     POINT AND THE SEAT ATTACHMENT POINT                      *
C  *                                                              *
C  ****************************************************************
C
C  COMPUTE THE SEAT CATAPULT ATTACHMENT POINT IN THE EARTH
C  SYSTEM (SAPE) ......
C
      CALL DIRCOS (DES,ESTIR)
      CALL TRANS (DSE,DES,3,3)
      CALL VECXYZ (SAPE,SAP,SRP,DSE,2)
C
C  COMPUTE THE AIRPLANE CATAPULT ATTACHMENT POINT IN THE EARTH
C  SYSTEM (AAPE) ......
C
      CALL DIRCOS (DEA,EAPIR)
      CALL TRANS (DAE,DEA,3,3)
      CALL VECXYZ (AAPE,AAP,XAP,DAE,2)
C
C  CALCULATE THE CATAPULT LENGTH COMPONENTS ......
C
      DO 30 I=1,3
 30   DXL(I) = SAPE(I) - AAPE(I)
C
C  DETERMINE THE DEFLECTED CATAPULT LENGTH ......
C
      CATL=SQRT(DXL(1)**2+DXL(2)**2+DXL(3)**2)
C
C  DETERMINE UNIT VECTOR ALONG THE CATAPULT EXTENSION ......
C
      DO 40 I=1,3
 40   IF(CATL.NE.0) CXUV(I) = DXL(I) / CATL
C
C  CALCULATE THE CATAPULT EXTENSION ......
C  (CORRECTING FOR CATAPULT DIRECTION DURING TRIM)
C
      FUDGE = 1
      IF(INST.EQ.31.AND.DXL(3)*UP*DAE(3,3).GT.0.0) FUDGE = -1.
      CEX = CATL - FUDGE * UCL
C
C  CALCULATE THE CATAPULT EXTENSION COMPONENTS ......
C
      DO 50 I=1,3
 50   EXT(I) = CEX * CXUV(I)
C
C
C  *******************************************************************
C  *                                                                 *
```

```
C  *      DETERMINE THE VARIABLES CALCULATED FROM THE             *
C  *      EARTH VELOCITIES OF THE AIRPLANE ATTACHMENT             *
C  *      POINT AND THE SEAT ATTACHMENT POINT                     *
C  *                                                              *
C  ****************************************************************
C
C  DETERMINE THE SEAT CATAPULT ATTACHMENT POINT VELOCITY COMPONENTS
C  IN THE EARTH SYSTEM (USAPE) ......
C
      CALL VELXYZ (USAPE,UST,SAP,WSTIR,DSE)
C
C  DETERMINE THE AIRPLANE CATAPULT ATTACHMENT POINT VELOCITY COMPONENTS
C  IN THE EARTH SYSTEM (UAAPE) ......
C
      CALL VELXYZ (UAAPE,UAP,AAP,WAPIR,DAE)
C
C  CALCULATE THE RELATIVE VELOCITY BETWEEN CATAPULT ENDS
C
      DO 60 I=1,3
 60   CDV(I) = USAPE(I) - UAAPE(I)
C
C  CALCULATE THE CATAPULT EXTENTION RATE (CV)
C
      CALL DOTPRD (CV,CDV,CXUV,3)
C
C  CALCLATE EXTENTION VELOCITY VECTOR
C
      DO 70 I=1,3
 70   CDV(I) = CV * CXUV(I)
C
C  ****************************************
C  *                                      *
C  *           CATAPULT LOGIC             *
C  *                                      *
C  ****************************************
C
C  BYPASS IF PRIOR TO CATAPULT IGNITION ......
C
      IF(SW.NE.1.) GO TO 90
C
C  COMPUTE THE EXPOSED THERMAL AREA OF THE CATAPULT CHAMBER ......
C
      THA = PC * (TLO + CEX*12.) + PA * 2.
C
C  COMPUTE THE FORCE DUE TO THE CATAPULT PRESSURE ......
C
      CALL CAD (CF,EF,EFDOT,IEF,EL,ELDOT,IEL,WK,WKDOT,IWK,WB,WBDOT,IWB,
     .          FL,TCP,TIME,CEX,CSK,CI,C,VI,PA,TF,CVH,CBP,C1,CV,C2,TI,
     .          THA,B,BXP,PT,R,TYPE,TSO,FSO,TDE)
      IF(FL.EQ.2.) FON = 1.
C
C  FIND THE EARTH SYSTEM COMPONENTS OF THE CATAPULT PRESSURE ......
C
      DO 80 I=1,3
 80   FCP(I) = -CF * CXUV(I)
C
C  ****************************************
C  *                                      *
```

```
C  *     CATAPULT STRUCTURAL SUPPORT       *
C  *                                       *
C  *****************************************
C
C  CHECK TO SEE IF THE CATAPULT MUST SUPPORT THE SEAT ......
C
      IF(CATL.GT.UCL) GO TO 120
C
C  FORCES DUE TO CATAPULT STRUCTURAL SPRING CONSTANT ......
C
 90   DO 100 I=1,3
 100  FSS(I) = SK * EXT(I)
C
C  FORCE DUE TO CATAPULT STRUCTURAL DAMPING ......
C
      DO 110 I=1,3
 110  FSD(I) = CK * CDV(I)
      GO TO 140
C
C  ZERO OUT THE CATAPULT STRUCTURAL FORCES AND MOMENTS WHEN
C  THE CATAPULT CAN SUPPORT THE SEAT ......
C
 120  DO 130 I=1,3
      FSD(I) = 0.
 130  FSS(I) = 0.
C
C  ********************************************
C  **********  TOTAL CATAPULT FORCES  **********
C  ********************************************
C
 140  DO 150 I=1,3
 150  FC(I) = FCP(I) + FSS(I) + FSD(I)
C
C  ***********************************************
C  *****  FORCES AND MOMENTS ON THE AIRPLANE  *****
C  ***********************************************
C
C  TRANSFORM THE EARTH SYSTEM FORCE COMPONENTS INTO THE
C  AIRPLANE BODY AXIS ......
C
      CALL MATMPY (FCA,DEA,FC,3,3,1)
C
C  CATAPULT MOMENTS ON THE AIRPLANE ......
C
      CALL CRSPRD (TCA,AAP,FCA)
C
C  ZERO THE FORCES AND TORQUES ACTING ON THE AIRPLANE IF SSFLG
C  IS EQUAL TO ZERO ......
C
      IF(SSFLG.NE.0) GO TO 160
      DO 155 I=1,3
 155  FCA(I) = TCA(I) = 0
C
C  *******************************************
C  *****  FORCES AND MOMENTS ON THE SEAT  *****
C  *******************************************
C
C  CATAPULT FORCES ON THE SEAT ......
```

```
C
 160  DO 165 I=1,3
 165  FC(I) = -FC(I)
C
C  TRANSFORM EARTH SYSTEM FORCE COMPONENTS INTO THE SEAT
C  BODY AXIS ......
C
      CALL MATMPY (FCS,DES,FC,3,3,1)
C
C  CATAPULT MOMENTS ON THE SEAT
C
      CALL CRSPRD (TCS,SAP,FCS)
C
 170  CONTINUE
C
      RETURN
      END
```

```
C
 160  DO 165 I=1,3
 165  FC(I) = -FC(I)
C
C  TRANSFORM EARTH SYSTEM FORCE COMPONENTS INTO THE SEAT
C  BODY AXIS ......
C
      CALL MATMPY (FCS,DES,FC,3,3,1)
C
C  CATAPULT MOMENTS ON THE SEAT
C
      CALL CRSPRD (TCS,SAP,FCS)
C
 170  CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE DR (TBF,
     .                 FDS,TDS,FDA,TDA,DLL,DBF,SW,
     .                 DAP,DBA,XAP,EAP,SRP,EST)
C
      COMMON /CICCAL/ ICCAL
      COMMON /COVRLY/ INST
      COMMON /CTIME/ TIME
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
C  ***************  DART TABLES  **************
C
C     TBF - DART BRAKING FORCE TABLE
C
C            THE INDEPENDENT VARIABLE IS THE LINE LENGTH (FT).
C            THE DEPENDENT VARIABLE IS THE BRAKING FORCE (LB).
C
C  ***************  DART OUTPUTS  ***************
C
C     FDS(3) - X,Y,Z BODY AXIS FORCE COMPONENTS ON THE SEAT (LB)
C     TDS(3) - X,Y,Z BODY AXIS MOMENT COMPONENTS ON THE SEAT (FT-LB)
C     FDA(3) - X,Y,Z BODY AXIS FORCE COMPONENTS ON THE AIRPLANE (FT)
C     TDA(3) - X,Y,Z BODY AXIS MOMENT COMPONENTS ON THE AIRPLANE(FT-LB)
C     DLL    - DISTANCE BETWEEN THE BRIDLE APEX AND THE AIRPLANE
C              ATTACHMENT POINT (FT)
C     DBF    - DART BRAKING FORCE (LB)
C     SW     - DART MODE FLAG
C                   0=PRIOR TO DART FORCE
C                   1=DART ON
C                   2=DART OFF
C
C  ***************  DART INPUTS  ***************
C
C     DAP(3) - X,Y,Z AIRPLANE BODY AXIS POSITON VECTOR OF
C              THE DART ATTACHMENT POINT (FT)
C     DBA(3) - X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE
C              DEPLOYED DART BRIDLE APEX (FT)
C     XAP(3) - X,Y,Z EARTH POSITION VECTOR OF THE AIRPLANE (FT)
C     EAP(3) - EARTH TO AIRPLANE EULER ANGLES (DEG)
C     SRP(3) - X,Y,Z EARTH POSITION VECTOR OF THE SEAT REFERENCE
C              POINT (FT)
C     EST(3) - EARTH TO SEAT EULER ANGLES (DEG)
C
C ///////////////////////////////////////////////////////////
C
      DIMENSION TBF(5),FDS(3),TDS(3),FDA(3),TDA(3),DAP(3),DBA(3),
     .          XAP(3),EAP(3),SRP(3),EST(3)
      DIMENSION DSE(3,3),DES(3,3),DAE(3,3),DEA(3,3),
     .          DAPE(3),DBAE(3),DELTA(3),DC(3),DF(3),
     .          ESTIR(3),EAPIR(3)
      DATA RPD /.01745329/
C
C  **************************
C  *****  INITIALIZATION  *****
C  **************************
C
      IF(ICCAL.NE.1) GO TO 20
      SW = 0
      DLL = DBF = 0
```

```
C
C  ////////////////////////////////////////////////////////////
C
C  ZERO OUT THE DART FORCES ......
C
 20    DO 30 I=1,3
 30    FDS(I) = TDS(I) = FDA(I) = TDA(I) = 0
C
C  BYPASS COMPONENT DURING STEADY STATE OR IF THE DART IS OFF ......
C
       IF(INST.EQ.31 .OR. SW.EQ.2.) GO TO 100
C
C  CONVERT EULER ANGLES FROM DEGREES TO RADIANS
C
       DO 40 I=1,3
       ESTIR(I) = EST(I) * RPD
 40    EAPIR(I) = EAP(I) * RPD
C
C  COMPUTE THE DIRECTION COSINE MATRICIES ......
C
       CALL DIRCOS (DES,ESTIR)
       CALL TRANS (DSE,DES,3,3)
       CALL DIRCOS (DEA,EAPIR)
       CALL TRANS (DAE,DEA,3,3)
C
C  EARTH AXIS POSITION OF THE AIRPLANE DART LINE ATTACHMENT
C  POINT ......
C
       CALL VECXYZ (DAPE,DAP,XAP,DAE,2)
C
C  EARTH AXIS POSITION OF THE DEPLOYED DART BRIDLE APEX ......
C
       CALL VECXYZ (DBAE,DBA,SRP,DSE,2)
C
C  CALCULATE THE DART LINE LENGTH ......
C
       DO 50 I=1,3
 50    DELTA(I) = DAPE(I) - DBAE(I)
       DLL = SQRT (DELTA(1)**2 + DELTA(2)**2 + DELTA(3)**2)
C
C  DETERMINE THE DART BRAKING FORCE ......
C
       NTBF = IBF(2)
       IF(DLL .LT. TBF(4)) GO TO 100
       IF(DLL .LT. TBF(3+NTBF)) GO TO 60
       IF(ICCAL.NE.1) SW = 2.
       IF(INST.EQ.26) WRITE(6,55) TIME
 55    FORMAT(/5X,*DART OFF AT TIME = *,F10.4,* SEC*/)
       GO TO 20
 60    IF(INST.EQ.26 .AND. SW.EQ.0.) WRITE(6,65) TIME
 65    FORMAT(/5X,*DART ON AT TIME = *,F10.4,* SEC*/)
       IF(ICCAL.NE.1) SW = 1.
       DBF = TBLU1(DLL,TBF(4),TBF(NTBF+4),1,-NTBF)
C
C  CALCULATE THE DIRECTION COSINES OF THE DART LINE ......
C
       DO 70 I=1,3
 70    DC(I) = DELTA(I)/DLL
```

```
C
C  EARTH COMPONENTS OF THE DART LINE LOAD ON THE SEAT ......
C
      DO 80 I=1,3
 80   DF(I) = DBF * DC(I)
C
C  ***************  SEAT FORCES AND MOMENTS  ***************
C
C  BODY AXIS FORCE COMPONENTS ON THE SEAT ......
C
      CALL MATMPY (FDS,DES,DF,3,3,1)
C
C  BODY AXIS MOMENT COMPONENTS ON THE SEAT ......
C
      CALL CRSPRD (TDS,DBA,FDS)
C
C  *************** AIRPLANE FORCES AND MOMENTS  ***************
C
C  BODY AXIS FORCE COMPONENTS OF THE AIRPLANE ......
C
      DO 90 I=1,3
 90   DF(I) = -DF(I)
      CALL MATMPY (FDA,DEA,DF,3,3,1)
C
C  BODY AXIS MOMENT COMPONENTS ON THE AIRPLANE ......
C
      CALL CRSPRD (TDA,DAP,FDA)
C
 100  RETURN
      END
```

```
      SUBROUTINE GP (TMF,
     .               FL,FMT,FST,TST,FPP,TPP,TIN,TLA,FSO,TSO,FPO,
     .               TPO,TRM,
     .               SW,UV,XMO,XYZ,EA,XR,XD,ER,ED,TDE,SRP,UST,EST,WST,
     .               XPP,UPP,EPP,WPP)
C
C
C  ***************  GP TABLES  ***************
C
C     TMF  -  PARACHUTE MORTAR FORCE TABLE
C
C              THE INDEPENDENT VARIABLE IS TIME (SEC)
C              THE DEPENDENT VARIABLE IS THE MORTAR FORCE (LB)
C
C  ***************  GP OUTPUTS  ***************
C
C     FL     - MORTAR MODE FLAG
C                   0 = PRIOR TO INITIATION
C                   1 = INITIATION UP TO LAUNCH
C                   2 = PARACHUTE LAUNCH
C                   3 = FORCES AND TORQUES OFF
C     FMT    - PARACHUTE MORTOR FORCE MAGNITUDE (LB)
C     FST(3) - X,Y,Z SEAT BODY AXIS FORCE VECTOR ACTING
C              ON THE SEAT (LB)
C     TST(3) - X,Y,Z SEAT BODY AXIS TORQUE VECTOR ACTING
C              ON THE SEAT (FT/LB)
C     FPP(3) - X,Y,Z EARTH SYSTEM FORCE VECTOR ACTING ON THE
C              PARACHUTE PACK (LB)
C     TPP(3) - X,Y,Z PARACHUTE PACK BODY AXIS TORQUE VECTOR ACTING
C              ON THE PARACHUTE PACK (FT-LB)
C     TIN    - PARACHUTE MORTAR INITIATION TIME (SEC)
C     TLA    - PARACHUTE LAUNCH TIME (SEC)
C     FSO(3) - X,Y,Z SEAT BODY AXIS FORCE COMPONENTS EXERTED ON
C              THE SEAT AT STRIPOFF (LB)
C     TSO(3) - X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS EXERTED ON
C              THE SEAT AT STRIPOFF (FT-LB)
C     FPO(3) - X,Y,Z PARACHUTE PACK BODY AXIS FORCE COMPONENTS
C              EXERTED ON THE SEAT AT STRIPOFF (LB)
C     TPO(3) - X,Y,Z PARACHUTE PACK BODY AXIS TORQUE COMPONENTS
C              EXERTED ON THE PACK AT STRIPOFF (FT-LB)
C     TRM(3) - X,Y,Z SEAT EARTH VELOCITY COMPONENTS TO PASS TO THE
C              PARACHUTE COMPONENT DURING TRIM (FT/SEC)
C
C  ***************  GP INPUTS  ***************
C
C     SW     - FLAG TO INITIATE THE MORTAR (1 = ON)
C     UV(3)  - X,Y,Z SEAT BODY AXIS MORTAR FORCE UNIT VECTOR
C     XMO(3) - X,Y,Z SEAT BODY AXIS LINEAR POSITION
C              VECTOR OF THE PARACHUTE DEPLOYMENT IMPULSE
C              MOMENT ARM (FT)
C     XYZ(3) - X,Y,Z SEAT BODY AXIS LINEAR POSITON VECTOR
C              OF THE PARACHUTE PACK (FT)
C     EA(3)  - SEAT TO PARACHUTE PACK EULER ANGLES (DEG)
C     XR     - PARACHUTE SHELF LINEAR SPRING CONSTANT (LB/FT)
C     XD     - PARACHUTE SHELF LINEAR DAMPING CONSTANT (LB/FT/SEC)
C     ER(3)  - X,Y,Z PARACHUTE SHELF ANGULAR SPRING CONSTANTS
C              (FT-LB/DEG)
C     ED(3)  - X,Y,Z PARACHUTE SHELF ANGULAR DAMPING CONSTANTS
```

```
C              (FT-LB/DEG/SEC)
C      TDE     - TIME DURATION FOR THE FORCES AND TORQUES TO DECAY TO
C                ZERO AFTER PARACHUTE LAUNCH (SEC)
C      SRP(3) - X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE SEAT (FT)
C      UST(3) - X,Y,Z SEAT BODY AXIS LINEAR VELOCITY VECTOR OF THE
C                SEAT (FT/SEC)
C      EST(3) - EARTH TO SEAT EULER ANGLES (DEG)
C      WST(3) - X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR
C                OF THE SEAT (DEG/SEC)
C      XPP(3) - X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE
C                PARACHUTE PACK (FT)
C      UPP(3) - X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF
C                THE PARACHUTE PACK (FT/SEC)
C      EPP(3) - EARTH TO PARACHUTE PACK EULER ANGLES (DEG)
C      WPP(3) - X,Y,Z PARACHUTE PACK BODY AXIS ANGULAR VELCITY VECTOR
C                OF THE PARACHUTE PACK (DEG/SEC)
C
C  DIMENSIONS OF CALLING ARGUMENTS ......
C
      DIMENSION TMF(5),FST(3),TST(3),FPP(3),TPP(3),TRM(3),
     .           UV(3),XMO(3),XYZ(3),EA(3),ER(3),ED(3),SRP(3),UST(3),
     .           EST(3),WST(3),XPP(3),UPP(3),EPP(3),WPP(3),
     .           FSO(3),TSO(3),FPO(3),TPO(3)
C
C  INTERNAL DIMENSIONS ......
C
      DIMENSION ESTIR(3),EPPIR(3),WSTIR(3),WPPIR(3),DES(3,3),
     .           DEST(3,3),DEP(3,3),DEPT(3,3),DSP(3,3),
     .           XS(3),DELTAX(3),SPRING(3),UXSE(3),
     .           DELTAV(3),RVEL(3),DAMP(3),FMORT(3),TMORT(3),
     .           PROJ(3),TORQUE(3),ANG(3),WSTE(3),WPPE(3),
     .           EAIR(3),DCEA(3,3),DCEAT(3,3),TEMP(3)
C
      COMMON /CTIME/ TIME
      COMMON /CICCAL/ ICCAL
      COMMON /COVRLY/ INST
      COMMON /CSSFLG/ SSFLG
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
      DATA RPD,DPR / .01745329, 57.29578 /
C
C  ***************************
C  *****  INITIALIZATION  *****
C  ***************************
C
      IF(ICCAL.NE.1) GO TO 10
C
      DO 2 I=1,3
  2   EAIR(I) = EA(I) * RPD
      CALL DIRCOS (DCEA,EAIR)
      CALL TRANS (DCEAT,DCEA,3,3)
      IF(TDE.EQ.0.99999) TDE = 0
      FL = FMT = TIN = TLA = TIMOR = 0
      DO 5 I=1,3
  5   TRM(I) = FSO(I) = TSO(I) = FPO(I) = TPO(I) = 0
      TYPE = 3HGUN
C
C  /////////////////////////////////////////////////////////////////
```

```
C
C   BYPASS CALCULATIONS IF THE PARACHUTE PACK HAS BEEN
C   RELEASED AND THE FORCES AND TORQUES HAVE DECAYED ......
C
 10    IF(FL.EQ.3.) GO TO 250
C
C   FACTOR FORCES AND TORQUES TO ZERO AFTER STRIPOFF .....
C
       IF(FL.NE.2.) GO TO 25
       TOFF = TLA + TDE
       DELTA = TOFF - TIME
       FACTOR = DELTA/TDE
       IF(DELTA.LE.0) FL = 3.
       IF(FL.EQ.3.) FACTOR = 0
       DO 20 I=1,3
       FST(I) = FSO(I) * FACTOR
       TST(I) = TSO(I) * FACTOR
       FPP(I) = FPO(I) * FACTOR
 20    TPP(I) = TPO(I) * FACTOR
       GO TO 250
C
C   SET THE TMORT AND FMORT VECTORS TO ZERO ......
C
 25    DO 30 I=1,3
       TMORT(I) = 0
 30    FMORT(I) = 0
       NMT = IMF(2)
C
C   *****  CHANGE FROM DEGREES TO RADIANS  *****
C
       DO 35 I=1,3
       ESTIR(I) = EST(I) * RPD
       WSTIR(I) = WST(I) * RPD
       EPPIR(I) = EPP(I) * RPD
 35    WPPIR(I) = WPP(I) * RPD
C
C   *****  CALCULATE THE DIRECTION COSINE MATRICES  *****
C
C   CALCULATE THE EARTH TO SEAT MATRIX ......
C
       CALL DIRCOS (DES,ESTIR)
C
C   CALCULATE THE SEAT TO EARTH MATRIX ......
C
       CALL TRANS (DEST,DES,3,3)
C
C   CALCULATE THE EARTH TO PARACHUTE PACK MATRIX ......
C
       CALL DIRCOS (DEP,EPPIR)
C
C   CALCULATE THE PARACHUTE PACK TO EARTH MATRIX ......
C
       CALL TRANS (DEPT,DEP,3,3)
C
C   CALCULATE THE SEAT TO PARACHUTE PACK MATRIX ......
C
       CALL MATMPY (DSP,DEP,DEST,3,3,3)
C
```

```
C   ***********************************************
C   *****  FORCES DUE TO LINEAR DISPLACEMENT   *****
C   ***********************************************
C
C   ---------  LINEAR SPRING FORCES  ---------
C
C   CALCULATE THE PARACHUTE PACK LINEAR POSITION VECTOR IN THE
C   SEAT COORDINATE SYSTEM  ......
C
      CALL VECXYZ (XS,XPP,SRP,DES,1)
C
C   DETERMINE THE LINEAR DISPLACEMENT FROM THE ATTACHMENT POINT,
C   AND CALCULATE THE SPRING FORCES IN THE SEAT SYSTEM ACTING ON
C   THE SEAT ......
C
      DO 40 I=1,3
      DELTAX(I) = XS(I) - XYZ(I)
 40   SPRING(I) = DELTAX(I) * XR
C
C   ----------  LINEAR DAMPING FORCES  ----------
C
C   DETERMINE THE EARTH VELOCITY OF THE POSITION THE PARACHUTE PACK
C   OCCUPIES IN THE SEAT COORDINATE SYSTEM ......
C
      CALL VELXYZ (UXSE,UST,XS,WSTIR,DEST)
C
C   DETERMINE THE RELATIVE VELOCITY WRT THE EARTH FRAME ......
C
      DO 45 I=1,3
 45   DELTAV(I) = UPP(I) - UXSE(I)
C
C   TRASFORM THIS DIFFERENCE INTO THE SEAT SYSTEM ......
C
      CALL MATMPY (RVEL,DES,DELTAV,3,3,1)
C
C   COMPUTE THE DAMPING FORCE ACTING ON THE SEAT ......
C
      DO 50 I=1,3
 50   DAMP(I) = RVEL(I) * XD
C
C   -----  SUM THE SPRING AND DAMPING FORCES ACTING ON THE SEAT  -----
C
      DO 60 I=1,3
 60   FST(I) = SPRING(I) + DAMP(I)
C
C   *******************
C   **  MORTAR LOGIC  **
C   ********************
C
      IF(SW.NE.1.) GO TO 130
C
      IF(FL.NE.0) GO TO 80
      IF(INST.EQ.26) WRITE(6,70) TYPE,TIME
 70   FORMAT(/5X,A8,* IGNITION AT TIME = *,F10.4,* SEC*/)
      TIN = TIME
      FL = 1.
C
C   CALCULATE THE MORTAR FORCE ......
```

```
C
 80   TIMOR = TIME - TIN
      FMT = TBLU1 (TIMOR,IMF(4),TMF(NMT+4),1,-NMT)
C
C  CALCULATE THE SEAT BODY AXIS MORTAR FORCE COMPONENTS
C  ACTING ON THE SEAT ......
C
      DO 90 I=1,3
 90   FMORT(I) = -1. * FMT * UV(I)
C
C  CALCULATE THE TORQUE ON THE SEAT FROM THE MORTAR ......
C
      CALL CRSPRD (TMORT,XMO,FMORT)
C
C  DOT THE LINEAR SPRING FORCES ONTO THE MORTAR UNIT VECTOR ......
C
      CALL DOTPRD (DOT,SPRING,UV,3)
C
C  IF THE SIGN OF THE DOT PRODUCT IS NEGATIVE, RETAIN THE SHELF FORCE ......
C
      IF(DOT.LE.0) GO TO 130
C
C  DOT THE TOTAL LINEAR RESTRAINT FORCE ONTO THE UNIT VECTOR ......
C
      CALL DOTPRD (DOT,FST,UV,3)
C
C  DETERMINE THE VECTOR COMPONENTS OF THE PROJECTION OF THE
C  RESTRAINT FORCE ONTO THE UNIT VECTOR ......
C
      DO 100 I=1,3
 100  PROJ(I) = DOT * UV(I)
C
C  DETERMINE THE FORCE VECTOR NORMAL TO THE UNIT VECTOR ......
C
      DO 110 I=1,3
 110  FST(I) = FST(I) - PROJ(I)
C
C  *********************************************************
C
C  DETERMINE THE TORQUE ON THE SEAT FROM THE RESTRAINTS ......
C
 130  CALL CRSPRD (TORQUE,XS,FST)
C
C  CALCULATE THE TOTAL FORCE ACTING ON THE SEAT ......
C
      DO 140 I=1,3
 140  FST(I) = FST(I) + FMORT(I)
C
C  CALCULATE THE FORCES ACTING ON THE PARACHUTE PACK IN THE
C  EARTH SYSTEM ......
C
      CALL MATMPY (FPP,DEST,FST,3,3,1)
      DO 150 I=1,3
 150  FPP(I) = -FPP(I)
C
C  *********************************************
C  *****  TORQUE DUE TO ANGULAR DISPLACEMENT  *****
C  *********************************************
```

```
C
C     -----  ANGULAR SPRING FORCES  -----
C
C     CALCULATE THE SEAT TO PARACHUTE PACK EULER ANGLES ......
C
          CALL COSDIR (ANG,DSP)
C
C     DETERMINE THE ANGULAR DISPLACEMENT FROM THE ATTACHMENT ANGLE,
C     AND CALCULATE THE SPRING COMPONENTS ACTING ON THE SEAT IN THE
C     ATTACHMENT AXIS SYSTEM ......
C
          DO 160 I=1,3
          DELTAX(I) = ANG(4-I)*DPR - EA(4-I)
  160     SPRING(I) = DELTAX(I) * ER(I)
C
C     -----  ANGULAR DAMPING FORCES  -----
C
C     DETERMINE THE ANGULAR VELOCITY OF THE PARACHUTE PACK IN THE
C     ATTACHMENT AXIS SYSTEM ......
C
          CALL MATMPY (WSTE,DEST,WST,3,3,1)
          CALL MATMPY (WPPE,DEPT,WPP,3,3,1)
          DO 170 I=1,3
  170     DELTAV(I) = WPPE(I) - WSTE(I)
          CALL MATMPY (TEMP,DES,DELTAV,3,3,1)
          CALL MATMPY (RVEL,DCEA,TEMP,3,3,1)
C
C     CALCULATE THE ANGULAR DAMPING TORQUE, AND SUM WITH THE ANGULAR
C     SPRING TORQUE ......
C
          DO 180 I=1,3
          DAMP(I) = RVEL(I) * ED(I)
  180     TEMP(I) = SPRING(I) + DAMP(I)
C
C     MOVE THE RESTRAINT TORQUES INTO THE SEAT SYSTEM ......
C
          CALL MATMPY (TST,DCEAT,TEMP,3,3,1)
C
C     CALCULATE THE BODY AXIS TORQUE CONSTANTS ACTING ON THE
C     PARACHUTE PACK ......
C
          CALL MATMPY (TPP,DSP,TST,3,3,1)
          DO 190 I=1,3
  190     TPP(I) = -TPP(I)
C
C     CALCULATE THE TOTAL MOMENT ON THE SEAT ......
C
          DO 200 I=1,3
  200     TST(I) = TST(I) + TMORT(I) + TORQUE(I)
C
C      IF THE MORTAR IS AT STRIPOFF ......
C
          IF(TIMOR.LT.TMF(NMT+3)) GO TO 225
          TLA = TIME
          FL = 2.
          IF(TDE.EQ.0) FL = 3.
          IF(FL.EQ.3.) GO TO 215
          DO 210 I=1,3
```

```
      FSO(I) = FST(I)
      TSO(I) = TST(I)
      FPO(I) = FPP(I)
 210  TPO(I) = TPP(I)
C
 215  IF(INST.EQ.26) WRITE(6,220) TYPE,TIME
 220  FORMAT(/5X,A6,* STRIPUFF AT TIME = *,F10.4,* SEC*/)
C
C  ZERO THE FORCES AND TORQUES ACTING ON THE SEAT IF SSFLG
C  IS EQUAL TO ZERO ......
C
 225  IF(SSFLG.NE.0) GO TO 240
      DO 230 I=1,3
 230  FST(I) = TST(I) = 0
C
C  SEND DATA TO PARACHTUE  PACK BODY TO ALLOW IT TO COMPUTE THE
C  SEAT EARTH VELOCITY DURING TRIM ......
C
 240  IF (INST.NE.31) GO TO 250
      CALL MATMPY (TRM,DEST,UST,3,3,1)
C
 250  RETURN
      END
```

```
      SUBROUTINE LI (TCW,
     .                 EC,ECD,IEC,TF,TFD,ITF,
     .                 FLA,SW1,FDO,TDO,FLP,FAP,VAP,FLL,ELM,ELC,DEM,
     .                 RMN,DIS,CON,TCG,UVL,RL,RLO,VL,VCG,PCG,CWT,TPE,PVL,
     .                 TLS,VLS,
     .                 OFF,BLI,APX,AP1,AP2,AP3,AP4,FTR,FSO,ULL,ULS,GOR,
     .                 TYP,FL,XDO,UDO,EDO,WDO,XPP,UPP,EPP,XPC,UPC)
C
C  DESIGNED BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C  THE EASIEST PARCHUTE LINE MODEL
C
C  ***************  LI TABLES  ***************
C
C     TCW    - STRETCHED PARACHUTE CANOPY WEIGHT TABLE
C
C                THE INDEPENDENT VARIABLE IS THE STRETCHED LENGTH (FT)
C                THE DEPENDENT VARIABLE IS THE STRETCHED WEIGHT (LB)
C
C  ***************  LI OUTPUTS  ***************
C
C  CREEP STRAIN IN PARACHUTE LINES
C
C     EC      - CREEP STRAIN IN PARACHUTE LINES (IN/IN)
C     ECD     - CREEP STRAIN RATE (IN/IN/SEC)
C     IEC     - INTEGRATION CONTROL
C
C  TIME DURATION OF PARACHUTE LINE LOAD (CHARACTERISTIC FUNTION)
C
C     TF      - TIME PARACHUTE LINES EXPERIENCE A NON-ZERO LOAD (SEC)
C     TFD     - RATE (EQUALS ONE WHEN LINES ARE UNDER LOAD, OTHERWISE ZERO)
C     ITF     - INTEGRATION CONTROL
C
C
C     FLA     - PARACHUTE PHASE
C                 0 = PRIOR TO INITIATION
C                 1 = INITIATION
C                 2 = LAUNCH
C                 3 = MORTAR OFF
C                 4 = LINESTRETCH
C                 5 = LINES SEVERED
C     SW1     - FLAG SET WHEN THE PARACHUTE IS BEHIND THE BRIDLE APEX
C     FDO(3)  - X,Y,Z DECELERATED OBJECT BODY AXIS FORCE COMPONENTS ACTING
C               ON THE DECELERATED OBJECT (LB)
C     TDO(3)  - X,Y,Z DECELERATED OBJECT BODY AXIS TORQUE COMPONENTS ACTING
C               ON THE DECELERATED OBJECT (FT-LB)
C     FLP(3)  - X,Y,Z FORCE COMPONENTS ACTING ON THE PARACHUTE (LB)
C               ( BODY AXIS FOR PACK  -  EARTH SYSTEM FOR CANOPY)
C     FAP(3)  - X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE
C               FORCE APPLICATION POINT (FT)
C     VAP(3)  - X,Y,Z EARTH SYSTEM VELOCITY COMPONENTS OF THE FORCE
C               APPLICATION POINT (FT/SEC)
C     FLL     - LINE LOAD (LB)
C     ELM     - MAXIMUM STRAIN EXPERIENCED BY THE PARACHUTE LINE
C               DURING ITS LOADING HISTORY (IN/IN)
C     ELC     - MAXIMUM STRAIN EXPERIENCED BY THE PARACHUTE LINE
C               DURING THE CURRENT LOADING CYCLE ONLY (IN/IN)
```

```
C      DEM      - MAXIMUM POSITIVE STRAIN RATE EXPERIENCED BY THE
C                 PARACHUTE LINE DURING ITS LOADING HISTORY (1/SEC)
C      RMN      - MAXIMUM NEGATIVE STRAIN RATE EXPERIENCED BY THE
C                 PARACHUTE LINE DURING THE CURRENT UNLOADING
C                 CYCLE ONLY (1/SEC)
C      DIS      - THE DISTANCE FROM THE ORIGIN OF THE DECELERATED OBJECT
C                 TO THE BRIDLE APEX (FT)
C      CON(4)   - COEFFICIENTS IN THE EQUATION FOR THE PLANE FORMED
C                 BY THE BRIDLE ATTACHMENT POINTS
C      TCG(20)  - PARACHUTE CENTER OF GRAVITY LOCATION ARRAY (FT)
C      UVL(3)   - PARACHUTE LINE UNIT VECTOR
C      RL       - PARACHUTE LINE LENGTH (FT)
C      RLO      - UNLOADED PARACHUTE LINE LENGTH (FT)
C      VL       - RATE OF CHANGE OF LINE LENGTH (FT/SEC)
C      VCG      - VELOCITY OF THE CANOPY CENTER OF GRAVITY ALONG THE
C                 PARACHUTE LINES (FT/SEC)
C      PCG      - STRETCHED CANOPY CENTER OF GRAVITY MEASURED ALONG THE
C                 PARACHUTE LINE FROM THE PARACHUTE PACK (FT)
C      CWT      - WEIGHT OF CANOPY PULLED FROM THE PARACHUTE PACK (LB)
C      TPE      - TYPE OF PARACHUTE (1=DRAG  2=RECOVERY)
C      PVL      - PREVIOUS TIMESTEP LINE VELOCITY (FT/SEC)
C      TLS      - TIME AT LINESTRETCH (SEC)
C      VLS      - RATE OF CHANGE OF LINE LENGTH AT LINESTRETCH (FT/SEC)
C
C  ***************  LI INPUTS  ***************
C
C      OFF     - FLAG TO SEVER LINES
C                     0 = LINES ATTACHED
C                     1 = LINES SEVERED
C      BLI     - NUMBER OF BRIDLE LINES
C      APX(3) - X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE
C                BRIDLE APEX (FT)
C      AP1(3) - X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE
C                FIRST BRIDLE LINE ATTACHMENT POINT (FT)
C      AP2(3) - X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE
C                SECOND BRIDLE LINE ATTACHMENT POINT (FT)
C      AP3(3) - X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE
C                THIRD BRIDLE LINE ATTACHMENT POINT (FT)
C      AP4(3) - X,Y,Z DECELERATED OBJECT BODY AXIS POSITION VECTOR OF THE
C                FOURTH BRIDLE LINE ATTACHMENT POINT (FT)
C      FTR     - PARACHUTE LINE MULTIPLICATION FACTOR
C      FSO     - CANOPY STRIPOUT FORCE (LB)
C      ULL     - PARACHUTE SUSPENSION LINE ULTIMATE LOAD (LB)
C      ULS     - PARACHUTE SUSPENSION LINE ULTIMATE STRAIN (IN/IN)
C      GOR     - NUMBER OF PARACHUTE GORES
C      TYP     - TYPE OF PARACHUTE (1=DRAG  2=RECOVERY)
C      FL      - MORTAR MODE FLAG
C                  0 = PRIOR TO INITIATION
C                  1 = INITIATION UP TO LAUNCH
C                  2 = PARACHUTE LAUNCH
C                  3 = MORTAR OFF
C                BODY (FT)
C      UDO(3) - X,Y,Z DECELERATED OBJECT BODY AXIS VELOCITY VECTOR
C                OF THE DECELERATED BODY (FT/SEC)
C      WDO(3) - X,Y,Z DECELERATED OBJECT BODY AXIS ANGULAR VELOCITY
C                COMPONENTS OF THE DECELERATED OBJECT (DEG/SEC)
C      XPP(3) - X,Y,Z EARTH FRAME POSITON VECTOR OF THE PARACHUTE
C                PACK (FT)
```

AD-A096 597 BOEING MILITARY AIRPLANE CO SEATTLE WA F/G 1/3
ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM. VOLUME --ETC(U)
SEP 80 C L WEST, B R UMMEL, R F YURCZYK F33615-79-C-3407
UNCLASSIFIED AFWAL-TR-80-3014-VOL-1 NL

5 OF 8

AD A
096597

```
C       UPP(3) - X,Y,Z PARACHUTE PACK EARTH SYSTEM VELOCITY
C                VECTOR (FT/SEC)
C       EPP(3) - EARTH TO PARACHUTE PACK EULER ANGLES (DEG)
C       XPC(3) - X,Y,Z EARTH SYSTEM POSITION VECTOR OF THE PARACHUTE
C                CANOPY (FT)
C       UPC(3) - X,Y,Z EARTH SYSTEM VECTOR VECTOR OF THE PARACHUTE
C                CANOPY (FT)
C
C
C   DIMENSION OF CALLING ARGUMENTS ......
C
        DIMENSION TCW(5),FDO(3),TDO(3),FLP(3),FAP(3),VAP(3),CON(4),
     .            TCG(20),UVL(3),APX(3),AP1(3),AP2(3),AP3(3),AP4(3),
     .            XDO(3),UDO(3),EDO(3),WDO(3),XPP(3),UPP(3),EPP(3),
     .            XPC(3),UPC(3)
C
C   INTERNAL DIMENSIONS ......
C
        DIMENSION WDOIR(3),EDOIR(3),DEO(3,3),DOE(3,3),XPPDO(3),
     .            UVV(3),FSTP(3),FADO(3),UPPPOS(3),
     .            UPPREL(3),UPPDO(3),EPPIR(3),FDOT(3),
     .            XPCS(3),XPCDO(3)
C
        COMMON /CTIME/ TIME
        COMMON /CICCAL/ICCAL
        COMMON /COVRLY/ INST
        COMMON /CPFLAG/ DUM,ITINC
        COMMON /CIO/ IREAD,IWRITE,IDIAG
C
        DATA RPD / .01745329 /
        DATA GRAV /32.174/
C
C   ****************************
C   *****  INITIALIZATION  *****
C   ****************************
C
        IF(ICCAL.NE.1) GO TO 70
C
C   MISC INITIALIZATION ......
C
        TPE = TYP
        FLA = SW1 = FLL = ELM = ELC = DEM = RMN = RL = RLO = 0
        VCG = PCG = CWT = PVL = TLS = VLS = 0
        IF(OFF .EQ. 0.99999) OFF = 0
        DO 10 I=1,3
        IF(APX(I) .EQ. 0.99999) APX(I) = 0
        IF(AP2(I) .EQ. 0.99999) AP2(I) = 0
        IF(AP3(I) .EQ. 0.99999) AP3(I) = 0
 10     IF(AP4(I) .EQ. 0.99999) AP4(I) = 0
C
C   CALCULATE THE DISTANCE FROM THE ORIGIN OF THE DECELERATED OBJECT
C   TO THE BRIDLE APEX ......
C
        DIS = SQRT ( APX(1)**2 + APX(2)**2 + APX(3)**2 )
C
C   CALCULATE THE CONSTANTS FOR THE EQUATION DEFINING THE
C   BRIDLE ATTACHMENT PLANE ......
C
```

```
      CON(1) = DET3(1.,AP1(2),AP1(3),1.,AP2(2),AP2(3),1.,AP3(2),AP3(3))
      CON(2) = DET3(AP1(1),1.,AP1(3),AP2(1),1.,AP2(3),AP3(1),1.,AP3(3))
      CON(3) = DET3(1.,AP1(1),AP1(2),1.,AP2(1),AP2(2),1.,AP3(1),AP3(2))
      CON(4) = DET3(AP1(2),AP1(1),AP1(3),AP2(2),AP2(1),AP2(3),
     .              AP3(2),AP3(1),AP3(3))
C
C  COMPUTE THE PARACHUTE CANOPY CG TABLE ......
C
      DO 15 I=1,20
 15   TCG(I) = 0
      WRITE(6,20)
 20   FORMAT(//5X,*--- STRETCHED CANOPY CG TABLE FOR COMPONENT*,
     .       * LI ---*,//18X,*LINE*,12X,*CG*//)
      NA = TCW(2)
      TOTALM = 0
      TOTALW = TCW(2*NA+3)
      DO 30 I=2,NA
      TOTALM = TOTALM + ((TCW(3+I)-TCW(2+I))/2.+TCW(2+I))*
     .        (TCW(NA+3+I)-TCW(NA+2+I))
      TCG(I) = (TOTALM + (TOTALW-TCW(NA+3+I))*TCW(3+I))/TOTALW
 30   TCG(I) = TCW(3+I) - TCG(I)
      WRITE(6,40) (TCW(I+3),TCG(I),I=1,NA)
 40   FORMAT(18X,F5.2,10X,F7.4)
C
      DO 60 I=1,3
      FDO(I) = 0
      TDO(I) = 0
      FLP(I) = 0
      FAP(I) = 0
      VAP(I) = 0
 60   UVL(I) = 0
C
C  ---  BYPASS THE COMPONENT IF FL DOES NOT EQUAL 2 OR FLA EQUALS 4  ---
C
 70   IF(FL.EQ.1.) FLA = 1.
      IF(FL.LE.1. .OR. FLA.EQ.5.) GO TO 330
C
      IF(TYP.EQ.1.) TYPE = 4HDRAG
      IF(TYP.EQ.2.) TYPE = 8HRECOVERY
C
C  ---  IF THE LINES HAVE BEEN SEVERED  ---
C
      IF(OFF.NE.1.) GO TO 100
      FLA = 5.
      FLL = 0
      DO 80 I=1,3
 80   FDO(I) = TDO(I) = FLP(I) = UVL(I) = VAP(I) = FAP(I) = 0
      IF(INST.EQ.26) WRITE(6,90) TYPE,TIME
 90   FORMAT(/5X,A8,* CHUTE LINES SEVERED AT TIME = *,F10.4,* SEC*/)
      GO TO 330
C
C  ---  CHANGE FROM DEGREES TO RADIANS  ---
C
 100  DO 110 I=1,3
      WDOIR(I) = WDO(I) * RPD
      EDOIR(I) = EDO(I) * RPD
 110  EPPIR(I) = EPP(I) * RPD
C
```

```
C  ----  CALCULATE DEO  ----
C
      CALL DIRCOS (DEO,EDDIR)
C
      IF(FLA.EQ.4.) GO TO 260
      FLA = 2.
C
C  ********************************
C  **                            **
C  **     PRIOR TO LINESTRETCH   **
C  **                            **
C  ********************************
C
      IF(FL.EQ.3.) FLA = 3.
C
C  ----  IF THE CHUTE IS INSIDE THE BRIDLE  ----
C
      IF(BLI.EQ.1.) GO TO 175
      IF(SW1.EQ.1.) GO TO 150
      CALL VECXYZ (XPPDO,XPP,XDO,DEO,1)
      IF(SQRT(XPPDO(1)**2+XPPDO(2)**2+XPPDO(3)**2).GE.DIS+1.) GO TO 140
C
C  CALCULATE THE EARTH SYSTEM VELOCITY OF THE PARACHUTE PACK
C  POSITION IN THE DECELERATED OBJECT COORDINATE SYSTEM ......
C
      CALL TRANS (DOE,DEO,3,3)
      CALL VELXYZ (UPPPOS,UDO,XPPDO,WDOIR,DOE)
C
C  COMPUTE THE RELATIVE VELOCITY OF THE PARACHUTE PACK WRT THE
C  DECELERATED OBJECT IN THE EARTH SYSTEM ......
C
      DO 120 I=1,3
 120  UPPREL(I) = UPPPOS(I) - UPP(I)
C
C  DETERMINE THE RELATIVE VELOCITY OF THE PARACHUTE PACK IN THE
C  DECELERATED OBJECT SYSTEM ......
C
      CALL MATMPY (UPPDO,DEO,UPPREL,3,3,1)
C
C  CALCULATE THE UNIT VECTOR OF UPPDO ......
C
      RESULT = SQRT(UPPDO(1)**2+UPPDO(2)**2+UPPDO(3)**2)
      DO 130 I=1,3
 130  UVV(I) = UPPDO(I)/RESULT
C
C  APPROXIMATE THE FORCE APPLICATION POINT FROM THE VELOCITY
C  VECTOR ......
C
      CALL LIBRIDL (FAP,
     .              APX,AP1,AP2,AP3,AP4,CON,BLI,UVV,XPPDO)
      GO TO 180
C
 140  SW1 = 1.
C
C  ----  CALCULATE THE FORCE APPLICATION POINT  ----
C
C  DETERMINE THE UNIT VECTOR FROM THE PARACHUTE PACK TO THE BRIDLE
C  APEX IN THE DECELERATED OBJECT COORDINATE SYSTEM ......
```

```
C
 150  CALL VECXYZ (XPPDO,XPP,XDO,DEO,1)
C
      DO 160 I=1,3
 160  UVL(I) = APX(I) - XPPDO(I)
C
      RESULT = SQRT(UVL(1)**2+UVL(2)**2+UVL(3)**2)
C
      DO 170 I=1,3
 170  UVL(I) = UVL(I)/RESULT
C
 175  CALL LIBRIDL (FAP,
     .                APX,AP1,AP2,AP3,AP4,CON,BLI,UVL,XPPDO)
C
C  ----  CALCULATE THE LINE VARIABLES  ----
C
 180  CALL LILINE (RL,UVL,VL,VAP,
     .              FAP,XDO,UDO,EDOIR,WDOIR,XPP,UPP,DEO)
C
C  ---- DETERMINE THE CANOPY CG POSITION AND WEIGHT  ----
C
      NA = TCW(2)
      PCG = TBLU1 (RL,TCW(4),TCG(1),1,-NA)
      CWT = TBLU1(RL,TCW(4),TCW(NA+4),1,-NA)
C
C  -----  CHECK FOR LINESTRETCH -----
C
      IF(RL.GE.TCW(NA+3)) GO TO 265
C
C  ----  CALCULATE THE CANOPY STRIPOUT FORCE  ----
C
      DO 190 I=1,3
 190  FSTP(I) = FSO * (-UVL(I))
C
C  ----  CALCULATE THE FORCE ACTING ON THE DECELERATED OBJECT RESULTING  ----
C              FROM PULLING THE PARACHUTE FROM THE PACK
C
      DO 200 I=2,NA
 200  IF(RL.LT.TCW(I+3)) GO TO 210
C
 210  DWDL = (TCW(NA+I+3)-TCW(NA+I+2))/(TCW(I+3)-TCW(I+2))
      DMDL = DWDL/GRAV
      MPDOT = DMDL * VL
C
      DO 220 I=1,3
 220  FADO(I) = MPDOT * (-UVL(I))
C
C  ----  SUM THE FORCES ACTING ON THE DECELERATED OBJECT  ----
C
      DO 230 I=1,3
 230  FDOT(I) = FSTP(I) + FADO(I)
      CALL MATMPY (FDO,DEO,FDOT,3,3,1)
C
C  ----  CALCULATE THE TORQUE ACTING ON THE DECELERATED OBJECT  ----
C
      CALL CRSPRD (TDO,FAP,FDO)
C
C  ----  SUM THE FORCES ACTING ON THE PARACHUTE PACK  ----
```

```
C
      DO 240 I=1,3
 240  FLP(I) = -FSTP(I)
C
C  -----  CALCULATE THE CANOPY CG VELOCITY ALONG THE PARACHUTE  ----
C         LINES WITH RESPECT TO THE FORCE APPLICATION POINT
C
      DO 250 I=2,NA
 250  IF(RL.LT.TCW(I+3)) GO TO 260
C
 260  DCGDL = (TCG(I)-TCG(I-1))/(TCW(I+3)-TCW(I+2))
      VCG = VL - VL * DCGDL
      GO TO 330
C
C  *******  AT LINESTRETCH  *******
C
 265  FLA = 4.
      TLS = TIME
      VCG = 0
      DO 270 I=1,3
      TDO(I) = FDO(I) = FLP(I) = 0
 270  CONTINUE
C
C  CALCULATE THE UNLOADED LINE LENGTH .....
C
      CALL VECXYZ (XPCS,XPC,XDO,DEO,1)
      RLO = SQRT((FAP(1)-XPCS(1))**2 + (FAP(2)-XPCS(2))**2 +
     .              (FAP(3)-XPCS(3))**2)
      RL = RLO
C
C  WRITE THE LINESTRETCH MESSAGE ......
C
      IF(INST.EQ.26) WRITE (6,275) TYPE,TIME
 275  FORMAT(/5X,A8,* CHUTE LINESTRETCH AT TIME = *,F10.4,* SEC*/)
C
C  ********************************
C  **                            **
C  **      AFTER LINESTRETCH     **
C  **                            **
C  ********************************
C
C
C  ----  CALCULATE THE FORCE APPLICATION POINT  ----
C
C  DETERMINE THE UNIT VECTOR FROM THE PARACHUTE CANOPY TO THE BRIDLE
C  APEX IN THE DECELERATED OBJECT COORDINATE SYSTEM ......
C
 280  IF(BL1.EQ.1.) GO TO 305
      CALL VECXYZ (XPCDO,XPC,XDO,DEO,1)
C
      DO 290 I=1,3
 290  UVL(I) = APX(I) - XPCDO(I)
C
      RESULT = SQRT(UVL(1)**2+UVL(2)**2+UVL(3)**2)
C
      DO 300 I=1,3
 300  UVL(I) = UVL(I)/RESULT
C
```

```
 305  CALL LIBRIDL (FAP,
     .                 APX,AP1,AP2,AP3,AP4,CON,BLI,UVL,XPCDO)
C
C  ----  CALCULATE THE LINE VARIABLES  ----
C
      CALL LILINE (RL,UVL,VL,VAP,
     .               FAP,XDO,UDO,EDOIR,WDOIR,XPC,UPC,DEO)
      IF(VLS.EQ.0) VLS = VL
C
C  ----  CALCULATE THE PARACHUTE LINE LOAD  ----
C
C  LOGIC TO DETERMINE THE LINE ACCELERATION .....
C
      AL = 0
      IF(INST.EQ.26) AL = VL - PVL
      IF(ITINC.EQ.1 .AND. INST.EQ.26) PVL = VL
C
      TALS = TIME - TL
      IF(TALS.LT.0.) TALS = 0.
      CALL LILOAD (FLL,FTR,EC,ECD,IEC,TF,TFD,ITF,
     .               TALS,AL,VL,RL,RLO,GOR,ULL,ULS,TYPE,
     .               ELM,ELC,DEM,RMN)
C
C  ----  CALCULATE THE FORCES AND TORQUES ACTING ON THE OBJECT  ----
C
      DO 310 I=1,3
 310  FDOT(I) = FLL * (-UVL(I))
      CALL MATMPY (FDO,DEO,FDOT,3,3,1)
      CALL CRSPRD (TDO,FAP,FDO)
C
C  ----  CALCULATE THE FORCES ACTING ON THE PARACHUTE CANOPY  ----
C
      DO 320 I=1,3
 320  FLP(I) = -FDOT(I)
C
 330  RETURN
      END
```

```
      SUBROUTINE LIBRIDL (FAP,
     *                    APX,AP1,AP2,AP3,AP4,CON,BLI,UV,XPDO)
C
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
C  *****  LIBRIDL OUTPUTS  *****
C
C     FAP(3)   - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION
C                VECTOR OF THE FORCE APPLICATION POINT (FT)
C
C  *****  LIBRIDL INPUTS  *****
C
C     APX(3)   - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C                OF THE BRIDLE APEX (FT)
C     AP1(3)   - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C                OF THE FIRST BRIDLE ATACHMENT POINT (FT)
C     AP2(3)   - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C                OF THE SECOND BRIDLE ATTACHMENT POINT (FT)
C     AP3(3)   - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C                OF THE THIRD BRIDLE ATTACHMENT POINT (FT)
C     AP4(3)   - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C                OF THE FOURTH BRIDLE ATTACHMENT POINT (FT)
C     CON(4)   - CONSTANTS IN THE EQUATION FOR A PLANE
C     BLI      - NUMBER OF BRIDLE LINES
C     UV(3)    - UNIT VECTOR FROM THE PARACHUTE PACK TO THE BRIDLE APEX
C                IN THE DECELERATED OBJECT COORDINATE SYSTEM
C     XPDO(3)  - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION
C                VECTOR OF THE PARACHUTE (FT)
C
C
      DIMENSION FAP(3),APX(3),AP1(3),AP2(3),AP3(3),AP4(3),
     *          CON(4),XI(3),UV(3),XPDO(3)
C
      GO TO (10,30,40,50),BLI
C
 10   DO 20 I=1,3
 20   FAP(I) = AP1(I)
      GO TO 60
C
 30   CALL BRIDL2 (FAP,APX,XPDO,AP1,AP2)
      GO TO 60
C
 40   CALL LINEPL (XI,CON,APX,UV)
      CALL BRIDL3 (FAP,APX,UV,XPDO,AP1,AP2,AP3,XI)
      GO TO 60
C
 50   CALL LINEPL (XI,CON,APX,UV)
      CALL BRIDL4 (FAP,APX,UV,XPDO,AP1,AP2,AP3,AP4,XI)
C
 60   RETURN
      END
```

```
      SUBROUTINE LILINE (RL,UVL,VL,VAP,
     .                   FAP,XDO,UDO,EDO,WDO,XPC,UPC,DEO)
C
C  *****  LILINE OUTPUTS  *****
C
C     RL      - DISTANCE FROM THE FORCE ATTACHMENT POINT TO THE
C               PARACHUTE CENTER OF GRAVITY (FT)
C     UVL(3)  - PARACHUTE LINE UNIT VECTOR
C     VL      - RATE OF CHANGE OF THE PARACHUTE LINE LENGTH (FT/SEC)
C     VAP(3)  - X,Y,Z EARTH SYSTEM VELOCITY VECTOR OF THE FORCE
C               APPLICATION POINT (FT/SEC)
C
C  *****  LILINE INPUTS  *****
C
C     FAP(3)  - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSTION VECTOR
C               OF THE FORCE APPLICATION POINT (FT)
C     XDO(3)  - X,Y,Z EARTH SYSTEM LINEAR POSITON VECTOR OF THE DECELERATED
C               OBJECT CENTER OF GRAVITY (FT)
C     UDO(3)  - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR VELOCITY VECTOR
C               OF THE DECELERATED OBJECT (FT/SEC)
C     EDO(3)  - EARTH TO DECELERATED OBJECT EULER ANGLES (RAD)
C     WDO(3)  - X,Y,Z DECELERATED OBJECT BODY AXIS ANGULAR VECLOCITY
C               VECTOR OF THE DECELERATED OBJECT (RAD/SEC)
C     XPC(3)  - X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE PARACHUTE (FT
C     UPC(3)  - X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE PARACHUTE
C               (FT/SEC)
C     DEO(3,3) - EARTH TO DECELERATED OBJECT DIRECTION COSINE MATRIX
C
      DIMENSION UVL(3),VAP(3),FAP(3),XDO(3),UDO(3),EDO(3),WDO(3),
     .          XPC(3),UPC(3),DEO(3,3),DOE(3,3),FAPE(3),DELTA(3)
C
C  **********  CALCULATE THE LINE LENGTH VARIABLES  **********
C
C  LOCATE THE FORCE APPLICATION POINT IN THE EARTH SYSTEM ....
C
      CALL TRANS (DOE,DEO,3,3)
      CALL VECXYZ (FAPE,FAP,XDO,DOE,2)
C
C  COMPUTE THE RESULTANTS AND DIRECTION COSINES
C
      DO 10 I=1,3
 10   DELTA(I) = FAPE(I) - XPC(I)
C
      RL = SQRT(DELTA(1)**2+DELTA(2)**2+DELTA(3)**2)
C
C  CALCULATE THE LINE UNIT VECTOR ......
C
      DO 20 I=1,3
 20   UVL(I) = DELTA(I)/RL
C
C  **********  CALCULATE THE LINE VELOCITY VARIABLES  **********
C
C  DETERMINE THE EARTH SYSTEM VELOCITY OF THE FAP ......
C
      CALL VELXYZ (VAP,UDO,FAP,WDO,DOE)
C
C  CALCULATE THE EARTH VELOCITY DIFFERENCE ......
C
```

```
      DO 30 I=1,3
 30   DELTA(I) = VAP(I) - UPC(I)
C
C  PROJECT THE DIFFERENCE ON THE PARACHUTE LINE ......
C
      CALL DOTPRD (VL,DELTA,UVL,3)
C
      RETURN
      END
```

```
      SUBROUTINE LILOAD (FTT,FCTR,EC,ECDOT,IEC,TF,TFD,ITF,
     .                   TALS,AX,VX,X,LO,GORES,ULTLD,ULTST,TYPE,
     .                   ELM,ELM1,DELOMAX,RMAXD2)
C
C ***** LILOAD OUTPUTS *****
C
C     FTT     - TENSILE LOAD (LB)
C     FCTR    - ULTIMATE STRENGTH MULTIPLICATION FACTOR
C     EC      - CREEP STRAIN IN TENSILE MEMBER (IN/IN)
C     ECDOT   - CREEP STRAIN RATE (1/SEC)
C     IEC     - INTEGRATION CONTROL FLAG
C     TF      - TIME DURATION OF PARACHUTE LINE LOAD (SEC)
C     TFD     - TIME DURATION RATE (EQUALS ONE UNDER LOAD)
C     ITF     - INTEGRATION CONTROL FLAG
C
C ***** LILOAD INPUTS *****
C
C     TALS     - TIME AFTER LINESTRETCH (SEC)
C     AX       - RATE OF CHANGE OF VX (FT/SEC/SEC)
C     VX       - RATE OF CHANGE OF THE LINE LENGTH (FT/SEC)
C     LO       - ORIGINAL UNSTRESSED LENGTH OF THE PARACHUTE LINES (FT)
C     GORES    - NUMBER OF PARACHUTE GORES
C     ULTLD    - ULTIMATE STRENGTH OF A PARACHUTE SUSPENSION
C                LINE (LB)
C     ULTST    - ULTIMATE STRAIN OF A PARACHUTE SUSPENSION
C                LINE (IN/IN)
C     TYPE     - ALPHANUMERIC FOR PARACHUTE TYPE
C
C ***** ARGUEMENTS INCLUDED TO SAVE VALUES *****
C
C     ELM      - MAXIMUM STRAIN EXPERIENCED BY THE TENSILE MEMBER (IN/IN)
C     ELM1     - MAXIMUM STRAIN EXPERIENCED DURING THE CURRENT
C                LOADING CYCLE (IN/IN)
C     DELOMAX  - THE MAXIMUM POSITIVE STRAIN RATE EXPERIENCED DURING
C                THE LOADING HISTORY (1/SEC)
C     RMAXD2   - THE MAXIMUM NEGATIVE STRAIN RATE EXPERIENCED DURING
C                THE CURRENT UNLOADING CYCLE (1/SEC)
C
C
C     JANUARY 1978 EDITION OF TENSILE LOAD-ELONGATION ANALOG FOR
C     MIL-C-5040E NYLON CORE-SLEEVE CORD . . .
C
C REFERENCE - AFFDL-TR-78-169  SIMULATION OF THE DYNAMIC TENSILE
C             CHARACTERISTICS OF NYLON PARACHUTE MATERIALS
C
C             AUTHOR - ROBERT E. MCCARTY  513-255-52516
C                      AFFDL/FER  W-PAFB, OHIO 45433
C
      DIMENSION EXA(6),EXD(6),EXC(5,3),TC(6),FC(3),CSR(6,3),
     . DPA(6),DPD(6),DPC(5,3),ELRLC(3),ELRRC(3),RATC(3)
C
      COMMON /COVRLY/ INST
      COMMON /CTIME/ TIME
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
      REAL KRL,KCR,KDP,L,LO
C
C ----- CREEP STRAIN RATE DATA FOR TABLE LOOK UP -----
```

```
C
C  TC IS AN INDEPENDENT VARIABLE ARRAY FOR TIME (SEC) ......
C
      DATA TC / 0., .2, .5, .9, 1.8, 20./
C
C  FC IS AN INDEPENDENT VARIABLE ARRAY FOR TENSILE LOAD (LB) ......
C
      DATA FC / 0., 80., 1000./
C
C  CSR IS A DEPENDENT ARRAY FOR CREEP STRAIN RATE (1/SEC) ......
C
      DATA CSR / 0., 0., 0., 0., 0., 0.,
     .           .06779, .03510, .01870, .01010, .001919, .001919,
     .           .06779, .03510, .01870, .01010, .001919, .001919 /
C
C  -----  MATERIAL UNLOADING CURVE FIT PARAMETERS  -----
C
C  DPA IS AN ARRAY OF ABSCISSAE FOR THE SIX FIXED KNOTS
C  IN A CUBIC SPLINE CURVE FIT USED TO REPRESENT MATERIAL
C  UNLOADING CHARACTERISTICS ......
C
      DATA DPA / 0.0, 0.072, 0.272, 0.65359, 0.73397, 1.0 /
C
C  DPO (LB) IS AN ARRAY OF ORDINATES FOR THE SIX FIXED KNOTS  ......
C
      DATA DPO / 0.67719, 15.140, 30.968, 17.945, 11.383, 0.0/
C
C  DPC IS AN ARRAY OF CUBIC SPLINE COEFFICIENTS ......
C
      DATA DPC / 91.851, 218.90, 2.3375, -80.664, -71.635,
     .           2778.2, -1013.6, -69.191, -148.32, 260.65,
     .           -17555.0, 1574.1, -69.126, 1696.0, -603.77/
C
C  -----  LOADING CURVE FIT PARAMETERS  -----
C
C  EXA (IN/IN) IS AN ARRAY OF ABSCISSAE FOR THE SIX FIXED KNOTS
C  IN THE CUBIC SPLINE CURVE FIT USED TO REPRESENT MATERIAL
C  LOADING CHARACTERISTICS ......
C
      DATA EXA /  0.0, .037226, .058852, .178888, .210448, .237515/
C
C  EXO (LB) IS AN ARRAY OF ORDINATES FOR THE SIX FIXED KNOTS ......
C
      DATA EXO / 0.0, 10.7213, 33.0281, 156.476, 205.580, 251.117/
C
C  EXC IS AN ARRAY OF CUBIC SPLINE COEFFICIENTS USED TO REPRESENT
C  THE MATERIAL LOADING CHARACTERISTICS ......
C
      DATA EXC / 122.991, 756.471, 1133.19, 1216.87, 2107.30,
     .          -3718.97, 20735.9, -3315.61, 4012.75, 24201.0,
     .           218974.0, -370731.0, 20350.3, 213226.0, -1484210. /
C
C  -----  PLASTIC STRAIN CHARACTERISTIC  -----
C
      DATA ELRLC / -.0508,.2178,3.5989/
      DATA ELRRC / -.0508,.2178,3.5989/
C
C  -----   DAMPING  STRAIN DEPENDENCE DATA  -----
```

```
C
      DATA RATC / -2.7208, 122.01, -272.36/
C
C  -----  MISC. DATA  -----
C
      DATA KRL,KCR,KDP,VSFDM / 3*1.0, 0.034 /
C
C  ***********************
C  *****  ELONGATION  *****
C  ***********************
C
C  EL (IN/IN) IS STRAIN BASED ON ORIGINAL UNSTRESSED LENGTH ......
C
      EL = FSW (X/LO-1., 0., 0., X/LO-1.)
      EL = EL * .237515/ULTST
C
C  ELO (IN/IN) IS THE STRAIN EXCLUDING CREEP STRAIN ......
C
      ELO = FSW (EL-EC, 0., 0., EL-EC)
C
C  ELM (IN/IN) IS THE MAXIMUM STRAIN EXPERIENCED DURING THE
C  LOADING HISTORY ......
C
      ELM = AMAX1(ELO,ELM)
C
C  ELM1 (IN/IN) IS THE MAXIMUM STRAIN EXPERIENCED DURING THE
C  CURRENT LOADING CYCLE ......
C
      ELM1 = FSW (VX, AMAX1(ELO,ELM1), ELO, ELO)
C
C  ELRL (IN/IN) IS THE UPPER BOUND FOR RESIDUAL STRESS ......
C
      ELRL = ((ELRLC(3)*ELM+ELRLC(2))*ELM+ELRLC(1))*ELM+.0018
C
C  ELRR (IN/IN) IS THE LOWER BOUND FOR RESIDUAL STRAIN ......
C
      ELRR = ((ELRRC(3)*ELM+ELRRC(2))*ELM+ELRRC(1))*ELM+.0018
C
C  TS (SEC) IS THE CUMULATIVE TIME FOR WHICH THE MEMBER EXPERIENCED
C  ZERO LOAD ......
C
      TS = TALS - TF
C
C  TSS IS THE RATIO OF TS TO THE VALUE OF RELAXATION TIME FOR
C  THE MATERIAL ......
C
      TSS = FSW ((TS/.3)-1., TS/.3, 1., 1.)
C
C  ELR (IN/IN) IS THE RESIDUAL STRAIN ......
C
      ELR = ELRL - TSS * ABS(ELRL-ELRR)
      ELR = RLIM (KRL*ELR, 0., KRL*ELR)
C
C  ELOT (IN/IN) IS THE LINEAR TRANSFORM OF STRAIN ......
C
      ELOT = (ELO-ELR)*ELM/(ELM-ELR+.00001)
C
C  ELS IS THE NORMALIZED STRAIN ......
```

```
C
      ELS = (ELM1-ELO)/(ELM1-ELR+.00001)
      ELS = FSW (ELS-1, ELS, ELS, 1.0)
      ELS = FSW (ELS, 0., 0., ELS)
C
C  L (FT) IS THE CURRENT UNSTRESSED LENGTH ......
C
      L = LO * (1.+ELR)
C
C  ELOT (IN/IN) IS THE LINEAR TRANSFORMATION OF STRAIN ......
C
      ELOT = RLIM(ELOT,0.,ELO)
C
C  DELO (1/SEC) IS THE STRAIN RATE BASED ON ORIGINAL UNSTRESSED
C  LENGTH ......
C
      DELO = VX/LO
C
C  DELOMAX (1/SEC) IS THE MAXIMUM POSITIVE STRAIN RATE EXPERIENCED
C  DURING THE LOADING HISTORY ......
C
      DELOMAX = AMAX1 (DELO, DELOMAX)
C
C  RMAXD1 (1/SEC) HAS THE VALUE OF DELO WHEN THE STRAIN RATE IS
C  NEGATIVE ......
C
      RMAXD1 = FSW (DELO, DELO, 0.00001, 0.00001)
C
C  RMAXD2 (1/SEC) IS THE MAXMIMUM NEGATIVE STRAIN RATE EXPERIENCED
C  DURING THE CURRENT UNLOADING CYCLE ......
C
      RMAXD2 = FSW(DELO,AMIN1(RMAXD1,RMAXD2),0.00001,0.00001)
C
C  CHECK TO SEE IF PARACHUTE LINES HAVE FAILED ......
C
      DO 10 I = 1,5
      IF(EXA(I).LE.ELO.AND.ELO.LT.EXA(I+1)) GO TO 30
 10   CONTINUE
      IF(INST.EQ.26) WRITE(6,20) TYPE,TIME
 20   FORMAT(/5X,A6,* CHUTE LINES FAILED AT TIME = *,F10.4,* SEC*,
     .               *    =====  RUN STOPPED  =====   */)
      STOP
C
C  **************************
C  *****  SPRING FORCE  *****
C  **************************
C
C  FSO (LB) IS THE TENSILE LOAD CALCULATE FROM THE CUBIC SPLINE
C  FIT ......
C
 30   D = ELO-EXA(I)
      FSO = ((EXC(I,3)*D+EXC(I,2))*D+EXC(I,1))*D+EXO(I)-EXO(1)
C
      DO 40 I=1,5
      IF(EXA(I).LE.ELOT.AND.ELOT.LT.EXA(I+1)) GO TO 50
 40   CONTINUE
C
C  FSR (LB) IS THE TENSILE LOAD CALCULATED FROM THE CUBIC SPLINE
```

```
C  FIT FOR THE MATERIAL REPEATED LOADING CHARACTERISTICS ......
C
 50    D = ELOT-EXA(I)
       FSR = ((EXC(I,3)*D+EXC(I,2))*D+EXC(I,1))*D+EXO(I)-EXO(1)
C
C  FSOL (LB) IS FSL LIMITED TO POSITIVE VALUES ......
C
       FSOL = RLIM (FSO, 0., FSO)
C
C  FSRL (LB) IS FSR LIMITED TO POSITIVE VALUES ......
C
       FSRL = RLIM (FSR, 0., FSR)
C
C  FS2 (LB) HAS THE VALUE OF FSOL FOR INITIAL LOADING AND
C  THE VALUE OF FSRL FOR REPEATED LOADING ......
C
       FS2 = FSW (ELO-ELM, FSRL, FSOL, FSOL )
C
C  FS1 IS THE SAME AS FS2, BUT IS ZERO WHEN THE LENGTH IS
C  LESS THAN THE CURRENT UNSTRESSED LENGTH ......
C
       FS1 = FSW(ELO-ELR,0.0,0.0,FS2)
C
C  FS (LB) IS THE CURRENT LOAD ......
C
       FS = FS1
C
C  ******************************
C  *****   DAMPING  EFFECT  *****
C  ******************************
C
C  RATIO IS A SCALAR QUANTITY USED TO ADJUST THE MAGNITUDE OF
C  LOAD ......
C
       RATIO = ((RATC(3)*ELM1+RATC(2))*ELM1+RATC(1))*ELM1
C
       DO 60 I=1,5
       IF(DPA(I).LE.ELS.AND.ELS.LT.DPA(I+1)) GO TO 70
 60    CONTINUE
C
C  FD4 (LB) IS THE LOAD CALCULATED FROM THE CUBIC SPLINE
C  FIT FOR THE MATERIAL UNLOADING CHARACTERISTIC ......
C
 70    D = ELS-DPA(I)
       FD4 = ((DPC(I,3)*D+DPC(I,2))*D+DPC(I,1))*D+DPO(I)
C
C  VSFD (SEC) IS THE LINEAR FUNCTION OF THE MAXIMUM STRAIN RATE ......
C
       VSFD = 0.90 + VSFDM * DELDMAX
C
C  FD3 (LB) IS THE VALUE OF FD4 SCALED FOR CURRENT CYCLE MAXIMUM
C  STRAIN AND MODIFIED BY A LINEAR VISCOUS DAMPING TERM ......
C
       FD3 = FD4*RATIO*KDP*VSFD*(ELM1-ELR)/(ELM-ELR+1.E-6)
       FD3 = FSW (FD3, 0., 0., FD3)
       FD3 = FSW (FD3-FS, FD3, FD3, FS)
C
C  FD1 (LB) IS THE SAME AS FD3 BUT LIMITED TO ZERO WHENEVER
```

```
C  THE STRAIN RATE IS ZERO OR POSITIVE ......
C
      FD1 = FSW(VX,FD3,0.,0.)
C
C  FD2 (LB) IS THE SAME AS FD1 EXCEPT THAT IT HAS THE VALUE ZERO
C  WHENEVER THE LENGTH IS LESS THAN THE CURRENT UNSTRESSED LENGTH ......
C
      FD2 = FSW(L-X,FD1,0.0,0.0)
C
C  FD (LB) IS THE CURRENT UNLOADING DECREMENT DERIVED FROM FD2 ......
C
      FACTOR = SQRT(DELO/RMAXD2)
      IF(AX.LE.10.) FACTOR = AX*(FACTOR-1.)/10. + 1.
      FD = FSW(AX,FD2,FD2,FACTOR*FD2)
C
C  *****  CALCULATE THE TENSILE LOAD (FT)  *****
C
      FT = FS - FD
      FTT = FT * GORES * FCTR * ULTLD/251.117
C
C  ***************************************************
C  *****  DETERMINE THE CURRENT CREEP STRAIN RATE  *****
C  ***************************************************
C
C  TF(SEC) IS THE CUMULATIVE TIME FOR WHICH TENSILE MEMBER
C  EXPERIENCED NONZERO LOAD ......
C
      TFD = 0.
      IF(FT.GT.0.0 .AND. ITF.NE.0) TFD = 1.
      IF(TF .GT. TC(6)) GO TO 80
C
C  DEC (1/SEC) IS THE CURRENT CREEP STRAIN RATE ......
C
      DEC = TBLU2 (TF,FT,TC,FC,CSR,1,1,-6,-3,6,3)
 80   IF(FT.LE.0.0) DEC = 0.0
      IF(TF.GT.TC(6)) DEC = 0.0
      IF(IEC.NE.0) ECDOT = DEC*KCR*1.8
C
      RETURN
      END
```

```
      SUBROUTINE LINDST (XYZ,R3I,DC,
     .                        PT1,PT2,PT3)
      DIMENSION XYZ(3),DC(3),PT1(3),PT2(3),PT3(3),DC12(3),DEL13(3)
C
C  THIS ROUTINE CALCULATES THE COORDINATES OF THE INTERSECTION
C  OF A NORMAL DRAWN FROM POINT THREE TO THE VECTOR PT1,PT2.
C  THE DIRECTION COSINES AND MAGNITUDE OF THE NORMAL ARE ALSO
C  CALCULATED.
C
C  *****  LINDST OUTPUTS  *****
C
C      XYZ(3) - X,Y,Z POSITION VECTOR OF THE INTERSECTION (FT)
C      R3I    - MAGNITUDE OF THE NORMAL VECTOR (FT)
C      DC(3)  - DIRECTION COSINES OF THE NORMAL VECTOR
C
C
C  DETERMINE THE MAGNITUDE OF VECTOR PT1,PT2.  DETERMINE ITS DIRECTION
C  COSINES ......
C
      R12=SQRT((PT1(1)-PT2(1))**2+(PT1(2)-PT2(2))**2+(PT1(3)-PT2(3))**2)
      DO 10 I=1,3
 10   DC12(I) = (PT2(I)-PT1(I))/R12
C
C  CALCULATE THE INTERSECTION POSITION VECTOR ......
C
      DO 15 I=1,3
 15   DEL13(I) = PT3(I) - PT1(I)
      CALL DOTPRD (R1I,DEL13,DC12,3)
C
      DO 20 I=1,3
 20   XYZ(I) = PT1(I) + R1I * DC12(I)
C
C  CALCULATE THE DIRECTION COSINES OF THE NORMAL ......
C
      R3I = SQRT ((XYZ(1)-PT3(1))**2 + (XYZ(2)-PT3(2))**2 +
     .             (XYZ(3)-PT3(3))**2)
      RMIN = .02 * R12
      IF (R3I - RMIN) 30,30,40
C
 30   DC(1) = DC(2) = DC(3) = 0
      GO TO 60
C
 40   DO 50 I=1,3
 50   DC(I) = (XYZ(I)-PT3(I))/R3I
C
 60   RETURN
      END
```

```
      SUBROUTINE LINEPL (X,C,XL,DC)
      DIMENSION X(3),C(4),XL(3),DC(3)
C
C  THIS ROUTINE DETERMINES THE COORDINATES OF THE INTERSECTION OF
C  A LINE AND A PLANE.
C
C  X(3) ARE THE COORDINATES OF THE INTERSECTION OF THE
C  LINE WITH THE PLANE.
C
C  THE PLANE IS DEFINED AS C(1)*X + C(2)*Y + C(3)*Z + C(4) = 0.
C
C  THE LINE IS DEFINED AS HAVING DIRECTION COSINES DC(3), PASSING
C  THROUGH A POINT WITH COORDINATES XL(3).
C
      DP=C(1)*DC(1)+C(2)*DC(2)+C(3)*DC(3)
      IF(DP.EQ.0.0)T=0
      IF(DP.NE.0.0)T=(-C(4)-C(1)*XL(1)-C(2)*XL(2)-C(3)*XL(3))/DP
      DO 10 I=1,3
 10   X(I) = XL(I) + T*DC(I)
      RETURN
      END
```

```
      SUBROUTINE LOOK(NN,R,VOUT)
C
C  ==============  CALLING ARGUMENTS  ==============
C
C      NN    - LOCATION IN R ARRAY OF DEPENDENT VARIABLE TABLE
C      R     - ARRAY CONTAINING AIRPLANE AERODYNAMIC TABLES
C      VOUT  - VALUE OF THE DEPENDENT VARIALBE DESIRED (OUTPUT)
C
C  ==============================================
C
      DIMENSION R(1),NIV(3),NSI(3),IND(3),NR(60)
      COMMON /REGIONS/
     1  NR1,  NR2,  NR3,  NR4,  NR5,  NR6,  NR7,  NR8,  NR9, NR10, NR11,
     2 NR12, NR13, NR14, NR15, NR16, NR17, NR18, NR19, NR20, NR21, NR22,
     3 NR23, NR24, NR25, NR26, NR27, NR28, NR29, NR30, NR31, NR32, NR33,
     4 NR34, NR35, NR36, NR37, NR38, NR39, NR40, NR41, NR42, NR43, NR44,
     5 NR45, NR46, NR47, NR48, NR49, NR50, NR51, NR52, NR53, NR54, NR55,
     6 NR56, NR57, NR58, NR59, NR60
C
      EQUIVALENCE (NIV(1),NIV1),(NIV(2),NIV2),(NIV(3),NIV3),
     1            (NSI(1),NSI1),(NSI(2),NSI2),(NSI(3),NSI3),
     2            (IND(1),IND1),(IND(2),IND2),(IND(3),IND3)
     3           ,(NR(1),NR1)
C
C  NUMBER OF INDEPENDENT VARIABLES ......
C
      NI = R(NN)
C
C  SET VOUT EQUAL TO ZERO IF THE NUMBER OF INDEPENDENT VARIABLES IS
C  ZERO ......
C
      IF(NI .NE. 0) GO TO 10
      VOUT = 0.
      GO TO 50
C
 10   K = NN + NI
      DO 20 I=1,NI
C
C  LOCATION OF INDEPENDENT VARIABLE TABLES ......
C
      NIT = R(NN+I)
      NRIT = NR(NIT)
C
C  NUMBER OF VALUES IN INDEPENDENT VARIABLE TABLE ......
C
      NIV(I) = R(NRIT)
C
C  LOCATION OF FIRST VALUE IN TABLE ......
C
      NSI(I) = NRIT + 1
C
C   LOCATION OF INDEPENDENT VARIABLE ......
C
      L = R(K+I) + .1
      IND(I) = L
 20   CONTINUE
C
C  LOCATION OF FIRST VALUE IN DEPENDENT VARIABLE TABLE ......
```

```
C
      ND = NN + 2*NI + 1
C
      IF(NI.EQ.2) GO TO 30
      IF(NI.EQ.3) GO TO 40
C
      VOUT = TBLU1(R(IND1),R(NSI1),R(ND),1,-NIV1)
      GO TO 50
 30   VOUT=TBLU2(R(IND1),R(IND2),R(NSI1),R(NSI2),R(ND),1,1,
     .           -NIV1,-NIV2,NIV1,NIV2)
      GO TO 50
 40   CALL TBLU3 (R(IND1),R(IND2),R(IND3),R(NSI1),R(NSI2),R(NSI3),
     .                R(ND),2,2,2,-NIV1,-NIV2,-NIV3,NIV1,NIV2,NIV3)
C
 50   RETURN
      END
```

```
      SUBROUTINE MP (TMP,
     .               EF,EFDOT,IEF,EL,ELDOT,IEL,WK,WKDOT,IWK,
     .               WB,WBDOT,IWB,
     .               FL,FST,TST,FPP,TPP,FM,EXM,VM,TLO,PC,R,CVH,
     .               TSO,FSO,TRM,
     .               SW,XYZ,EA,XR,XD,ER,ED,UV,CSK,VI,PA,PT,CBP,C,
     .               CI,PMW,GAM,TF,C1,C2,B,BXP,TI,TDE,SRP,UST,EST,WST,
     .               XPP,UPP,EPP,WPP)
C
C  DESIGNED BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C  THE EASIEST PARACHUTE MORTAR COMPONENT
C
C  ***************  MP TABLES  ***************
C
C     TMP - MORTAR PROPELLANT CONSUMPTION TABLE
C
C            THE INDEPENDENT VARIABLE IS THE PROPELLANT
C            WEB CONSUMED (IN) AND THE DEPENDENT VARIABLE
C            IS THE PROPELLANT CONSUMED (SLUGS)
C
C  ***************  MP OUTPUTS  ***************
C
C  INTERNAL FRICTION ENERGY ......
C
C     EF    - INTERNAL FRICTION ENERGY (FT-LB)
C     EFDOT - INTERNAL FRICTION ENERGY RATE (FT-LB/SEC)
C     IEF   - INTEGRATION CONTROL
C
C  HEAT LOSS ......
C
C     EL    - HEAT LOSS (FT-LB)
C     ELDOT - HEAT LOSS RATE (FT-LB/SEC)
C     IEL   - INTEGRATION CONTROL
C
C  MORTAR WORK ......
C
C     WK    - MORTAR WORK (FT-LB)
C     WKDOT - MORTAR WORK RATE (FT-LB/SEC)
C     IWK   - INTEGRATION CONTROL
C
C  PROPELLANT WEB BURNED ......
C
C     WB    - PROPELLANT WEB BURNED (IN)
C     WBDOT - PROPELLANT WEB BURN RATE (IN/SEC)
C     IWB   - INTEGERATION CONTROL
C
C     FL     - MORTAR MODE FLAG
C                0 = PRIOR TO INITIATION
C                1 = INITIATION UP TO LAUNCH
C                2 = PARACHUTE LAUNCH
C                3 = MORTAR OFF
C     FST(3) - X,Y,Z SEAT BODY AXIS FORCE COMPONENTS OF THE
C              MORTAR AND RESTRAINTS ON THE SEAT (LB)
C     TST(3) - X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS OF THE
C              MORTAR AND RESTRAINTS ON THE SEAT (FT-LB)
C     FPP(3) - X,Y,Z EARTH SYSTEM FORCE COMPONENTS OF THE
```

```
C                MORTAR AND RESTRAINTS ON THE PARACHUTE PACK (LB)
C     TPP(3) - X,Y,Z PARACHUTE PACK BODY AXIS TORQUE COMPONENTS OF THE
C                MORTAR AND RESTRAINTS ON THE PARACHUTE PACK (FT-LB)
C     FM     - MORTAR FORCE MAGNITUDE (LB)
C     EXM    - MORTAR EXTENSION (FT)
C     VM     - MORTAR EXTENSION VELOCITY (FT/SEC)
C     TLO    - INITIAL LENGTH OF THE MORTAR PRESSURE CHAMBER (IN)
C     PC     - CIRCUMFERENCE OF CATAPULT PRESSURE CHAMBER (IN)
C     R      - GAS CONSTANT (FT-LBF/SLUG-K)
C     CVH    - CONSTANT VOLUME SPECIFIC HEAT (FT-LBF/SLUG-K)
C     TSO    - MORTAR STRIPOFF TIME (SEC)
C     FSO    - FORCE AT MORTAR STRIPOFF (LB)
C     TRM(3) - X,Y,Z SEAT EARTH VELOCITY COMPONENTS TO PASS TO THE
C                PARACHUTE COMPONENT DURING TRIM (FT/SEC)
C
C  ***************  MP INPUTS  ***************
C
C     SW     - FLAG TO INITIATE MORTAR (1 = ON)
C     XYZ(3) - X,Y,Z SEAT BODY AXIS LINEAR POSITION VECTOR OF THE
C                PARACHUTE PACK ATTACHMENT POINT ON THE SEAT (FT)
C     EA(3)  - SEAT TO PARACHUTE PACK EULER ANGLES (DEG)
C     XK     - PARACHUTE SHELF LINEAR SPRING CONSTANT (LB/FT)
C     XD     - PARACHUTE SHELF LINEAR DAMPING CONSTANT (LB/FT/SEC)
C     ER(3)  - X,Y,Z PARACHUTE SHELF ANGULAR SPRING CONSTANT
C                (FT-LB/DEG)
C     ED(3)  - X,Y,Z PARACHUTE SHELF ANGULAR DAMPING CONSTANT
C                (FT-LB/DEG/SEC)
C     UV(3)  - X,Y,Z SEAT BODY AXIS MORTAR FORCE UNIT VECTOR
C     CSK    - MORTAR STROKE (FT)
C     VI     - INITIAL FREE VOLUME (IN**3)
C     PA     - PISTON AREA (IN**2)
C     PT     - TANG RELEASE PRESSURE (LB/IN**2)
C     CBP    - MORTAR BURST PRESSURE (LB/IN**2)
C     C      - MASS OF TOTAL PROPELLANT (SLUGS)
C     CI     - IGNITER PROPELLANT MASS (SLUGS)
C     PMW    - PROPELLANT MOLECULAR WEIGHT (LB/(LB-MOLE))
C     GAM    - RATIO OF SPECIFIC HEATS
C     TF     - CONSTANT VOLUME FLAME TEMPERATURE (DEG K)
C     C1     - FRICTION PROPORTIONALITY CONSTANT
C     C2     - HEAT LOSS CONSTANT
C     B      - BURN RATE PROPORTIONALITY CONSTANT (IN/SEC/(LB/IN**2))
C     BXP    - BURN RATE EXPONENT
C     TI     - MORTAR TEMPERATURE PRIOR TO IGNITION (DEG K)
C     TDE    - MORTAR FORCE DECAY TIME (SEC)
C     SRP(3) - X,Y,Z EARTH SYSTEM POSTION VECTOR OF THE
C                SEAT REFERENCE POINT (FT)
C     UST(3) - X,Y,Z SEAT BODY AXIS VELOCITY VECTOR OF THE
C                SEAT (FT/SEC)
C     EST(3) - EARTH TO SEAT EULER ANGLES (DEG)
C     WST(3) - X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY
C                OF THE SEAT (DEG/SEC)
C     XPP(3) - X,Y,Z EARTH SYSTEM POSITION VECTOR OF THE
C                PARACHUTE PACK (FT)
C     UPP(3) - X,Y,Z EARTH SYSTEM VELOCITY VECTOR OF
C                THE PARACHUTE PACK (FT/SEC)
C     EPP(3) - EARTH TO PARACHUTE PACK EULER ANGLES (DEG)
C     WPP(3) - X,Y,Z PARACHUTE PACK BODY AXIS ANGULAR VELOCITY
C                OF THE PARACHUTE PACK (DEG/SEC)
```

```
C
C   DIMENSIONS OF CALLING ARGUMENTS ......
C
      DIMENSION TMP(5),FST(3),TST(3),FPP(3),TPP(3),XYZ(3),EA(3),ER(3),
     .          ED(3),UV(3),SRP(3),UST(3),EST(3),WST(3),XPP(3),UPP(3),
     .          EPP(3),WPP(3),TRM(3)
C
C   INTERNAL DIMENSIONS ......
C
      DIMENSION DES(3,3),DEST(3,3),DEP(3,3),DEPT(3,3),DSP(3,3),
     .          DELTAX(3),DELTAV(3),ESTIR(3),WSTIR(3),EPPIR(3),
     .          WPPIR(3),XS(3),SPRING(3),UXSE(3),RVEL(3),
     .          DAMP(3),ANG(3),WSTE(3),WPPE(3),PROJ(3),
     .          FCAD(3),TORQUE(3),EAIR(3),DCEA(3,3),DCEAT(3,3),TEMP(3)
C
      COMMON / CTIME / TIME
      COMMON / CICCAL / ICCAL
      COMMON / COVRLY / INST
      COMMON / CSSFLG / SSFLG
      COMMON / CIO / IREAD,IWRITE,IDIAG
C
      DATA RPD,DPR / .01745329, 57.29578 /
C
C   ****************************
C   *****  INITIALIZATION  *****
C   ****************************
C
      IF(ICCAL.NE.1) GO TO 40
C
C   COMPUTE THE INITIAL LENGTH (TLO) AND CIRCUMFERENCE (PC) OF THE
C   MORTAR PRESSURE CHAMBER ......
C
      TLO = VI/PA
      PC = 2*SQRT(3.14159*PA)
C
C   COMPUTE THE CONSTANT VOLUME SPECIFIC HEAT (CVH) FOR THE MORTAR
C   PROPELLANT GIVEN THE GAS CONSTANT (GC) AND THE PROPELLANT
C   MOLECULAR WEIGHT (PMW) ......
C
      R = 89475.894/PMW
      CVH = R/(GAM-1.0)
C
      TYPE = 6HMORTAR
      VM = EXM = FM = FL = TSO = FSO = 0
      TRM(1) = TRM(2) = TRM(3) = 0
      IF(TOE .EQ. 0.99999) TOE = 0
      DO 30 I=1,3
   30 EAIR(I) = EA(I) * RPD
      CALL DIRCOS (DCEA,EAIR)
      CALL TRANS (DCEAT,DCEA,3,3)
C
C   ///////////////////////////////////////////////////////////
C
C   BYPASS THE REMAINING CODE IF THE MORTAR IS PAST THE
C   STRIPOFF ......
C
   40 IF(FL.EQ.3.) GO TO 260
C
```

```
C   CHANGE ANGULAR STATES FROM DEGREES TO RADIANS ......
C
       DO 50 I=1,3
       ESTIR(I) = EST(I) * RPD
       WSTIR(I) = WST(I) * RPD
       EPPIR(I) = EPP(I) * RPD
 50    WPPIR(I) = WPP(I) * RPD
C
C   CALCULATE THE EARTH TO SEAT MATRIX ......
C
       CALL DIRCOS (DES,ESTIR)
C
C   CALCULATE THE SEAT TO EARTH MATRIX ......
C
       CALL TRANS (DEST,DES,3,3)
C
C   CALCULATE THE EARTH TO PARACHUTE PACK MATRIX ......
C
       CALL DIRCOS (DEP,EPPIR)
C
C   CALCULATE THE PARACHUTE PACK TO EARTH MATRIX ......
C
       CALL TRANS (DEPT,DEP,3,3)
C
C   CALCULATE THE SEAT TO PARACHUTE PACK MATRIX ......
C
       CALL MATMPY (DSP,DEP,DEST,3,3,3)
C
C   ***********************************************************
C   *****  FORCES AND TORQUES DUE TO LINEAR DISPLACEMENT  *****
C   ***********************************************************
C
C   ----------  LINEAR SPRING FORCES  ----------
C
C   CALCULATE THE PARACHUTE PACK LINEAR POSITION VECTOR IN THE
C   SEAT COORDINATE SYSTEM ......
C
       CALL VECXYZ (XS,XPP,SKP,DES,1)
C
C   DETERMINE THE LINEAR DISPLACEMENT FROM THE ATTACHMENT POINT,
C   AND CALCULATE THE SPRING FORCES IN THE SEAT SYSTEM ACTING ON
C   THE SEAT ......
C
       DO 60 I=1,3
       DELTAX(I) = XS(I) - XYZ(I)
 60    SPRING(I) = DELTAX(I) * XR
C
C   ----------  LINEAR DAMPING FORCES  ----------
C
C   DETERMINE THE EARTH VELOCITY OF THE POSITION THE PARACHUTE PACK
C   OCCUPIES IN THE SEAT COORDINATE SYSTEM ......
C
       CALL VELXYZ (UXSE,UST,XS,WSTIR,DEST)
C
C   DETERMINE THE RELATIVE VELOCITY WRT THE EARTH SYSTEM ......
C
       DO 70 I=1,3
 70    DELTAV(I) = UPP(I) - UXSE(I)
```

```
C
C  TRANSFORM THIS DIFFERENCE INTO THE SEAT SYSTEM ......
C
      CALL MATMPY (RVEL,DES,DELTAV,3,3,1)
C
C  COMPUTE THE DAMPING FORCE ACTING ON THE SEAT ......
C
      DO 60 I=1,3
 60   DAMP(I) = RVEL(I) * XD
C
C  ----  SUM THE SPRING AND DAMPING FORCES ACTING ON THE SEAT  ---
C
      DO 90 I=1,3
 90   FST(I) = SPRING(I) + DAMP(I)
C
C  ************************
C  ***     MORTAR LOGIC    ***
C  ************************
C
      IF(SW.NE.1.) GO TO 170
C
C  CALCULATE THE MORTAR EXTENSION ......
C
      CALL DOTPRD (EXM,DELTAX,UV,3)
C
C  CALCULATE THE MORTAR EXTENSION VELOCITY ......
C
      CALL DOTPRD (VM,DELTAV,UV,3)
C
C  COMPUTE THE EXPOSED THERMAL AREA OF THE MORTAR CHAMBER ......
C
      THA = PC * (TLO + EXM*12.) + PA * 2.
C
C  COMPUTE THE FORCE DUE TO THE MORTAR PRESSURE ......
C
      CALL CAD (FM,EF,EFDOT,IEF,EL,ELDOT,IEL,WK,WKDOT,IWK,WB,WBDOT,IWB,
     .          FL,TMP,TIME,EXM,CSK,CI,C,VI,PA,TF,CVH,CBP,C1,VM,C2,TI,
     .          THA,B,BXP,PT,R,TYPE,TSO,FSO,TDE)
C
C  IF THE MORTAR IS AT STRIPOFF ......
C
      IF (FL.NE.3.) GO TO 120
      DO 110 I=1,3
 110  FST(I) = TST(I) = FPP(I) = TPP(I) = 0
      GO TO 260
C
C  CALCULATE THE SEAT BODY AXIS MORTAR FORCE COMPONENTS
C  ACTING ON THE SEAT ......
C
 120  DO 130 I=1,3
 130  FCAD(I) = -1. * FM * UV(I)
C
C  DOT THE LINEAR SPRING FORCE ONTO THE MORTAR UNIT VECTOR ......
C
      CALL DOTPRD (DOT,SPRING,UV,3)
C
C  IF THE SIGN OF THE DOT PRODUCT IS NEGATIVE, RETAIN THE SHELF FORCE  .....
C
```

```
      IF(DOT.LE.0) GO TO 155
C
C  DOT THE TOTAL LINEAR RESTRAINT FORCE ONTO THE UNIT VECTOR ......
C
      CALL DOTPRD (DOT,FST,UV,3)
C
C  CALCULATE THE COMPONENTS OF THIS PROJECTION ......
C
      DO 140 I=1,3
 140  PROJ(I) = DOT * UV(I)
C
C  DETERMINE THE FORCE VECTOR NORMAL TO THE UNIT VECTOR ......
C
      DO 150 I=1,3
 150  FST(I) = FST(I) - PROJ(I)
C
C  CALCULATE THE TOTAL FORCES AND MOMENTS ACTING ON THE SEAT ......
C
 155  DO 160 I=1,3
 160  FST(I) = FCAD(I) + FST(I)
C
C  ***************************************************
C
 170  CALL CRSPRD (TORQUE,XS,FST)
C
C  CALCULATE THE FORCES ACTING ON THE PARACHUTE PACK ......
C
      CALL MATMPY (FPP,DEST,FST,3,3,1)
      DO 180 I=1,3
 180  FPP(I) = -FPP(I)
C
C  ***********************************************
C  *****  TORQUE DUE TO ANGULAR DISPLACEMENT  *****
C  ***********************************************
C
C
C  ------   ANGULAR SPRING FORCES   -----
C
C  CALCULATE THE SEAT TO PARACHUTE PACK EULER ANGLES ......
C
      CALL COSDIR (ANG,DSP)
C
C  DETERMINE THE ANGULAR DISPLACEMENT FROM THE ATTACHMENT ANGLE,
C  AND CALCULATE THE SPRING COMPONENTS ACTING ON THE SEAT IN THE
C  ATTACHMENT AXIS SYSTEM ......
C
      DO 190 I=1,3
      DELTAX(I) = ANG(4-I)*DPR - EA(4-I)
 190  SPRING(I) = DELTAX(I) * ER(I)
C
C  -----  ANGULAR DAMPING FORCES  -----
C
C  CALCULATE THE BODY AXIS ANGULAR DAMPING CONSTANTS ACTING
C  ON THE SEAT IN THE ATTACHMENT AXIS SYSTEM ......
C
      CALL MATMPY (WSTE,DEST,WST,3,3,1)
      CALL MATMPY (WPPE,DEPT,WPP,3,3,1)
      DO 200 I=1,3
```

```
 200  DELTAV(I) = WPPE(I) - WSTE(I)
      CALL MATMPY (TEMP,DES,DELTAV,3,3,1)
      CALL MATMPY (VEL,DCEA,TEMP,3,3,1)
      DO 210 I=1,3
      DAMP(I) = RVEL(I) * ED(I)
 210  TEMP(I) = SPRING(I) + DAMP(I)
C
C  MOVE THE RESTRAINT TORQUES INTO THE SEAT SYSTEM ......
C
      CALL MATMPY (TST,DCEAT,TEMP,3,3,1)
C
C  CALCULATE THE BODY AXIS TORQUE CONSTANTS ACTING ON THE
C  PARACHUTE PACK ......
C
      CALL MATMPY (TPP,DSP,TST,3,3,1)
      DO 220 I=1,3
 220  TPP(I) = -TPP(I)
C
C  CALCULATE THE TOTAL MOMENT ON THE SEAT ......
C
      DO 230 I=1,3
 230  TST(I) = TST(I) + TORQUE(I)
C
C  ZERO THE FORCES AND TORQUES ACTING ON THE SEAT IF SSFLG
C  IS EQUAL TO ZERO ......
C
      IF(SSFLG.NE.0) GO TO 250
      DO 240 I=1,3
 240  FST(I) = TST(I) = 0
C
C  SEND DATA TO PARACHUTE PACK BODY TO ALLOW IT TO COMPUTE
C  SEAT EARTH VELOCITY DURING TRIM ......
C
 250  IF (INST.NE.31) GO TO 260
      CALL MATMPY (TRM,DEST,UST,3,3,1)
C
 260  RETURN
      END
```

```
      SUBROUTINE PAXIS (BMI,BPI,BM,BP,BMASS,DISP)
C
C  PARALLEL AXIS THEROEM FOR TRANSFERING THE MOMENTS AND
C  PRODUCTS OF INERTIA TO THE SEAT BODY AXIS
C
C  *****  CALLING PARAMETERS  *****
C
C  OUTPUT ......
C
C     BMI    - MASS MOMENT OF INERTIA WITH RESPECT TO THE SEAT
C              BODY AXIS (SLUG-FT**2)
C     BPI    - MASS PRODUCT OF INERTIA WITH RESPECT TO THE SEAT
C              BODY AXIS (SLUG-FT**2)
C
C  INPUT ......
C
C     BM     - MASS MOMENT OF INERTIA ABOUT THE BODY MASS CENTER
C              (SLUG-FT**2)
C     BP     - MASS PRODUCT OF INERTIA ABOUT THE BODY MASS CENTER
C              (SLUG-FT**2)
C     BMASS - BODY MASS (SLUGS)
C     DISP   - X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE BODY
C              MASS CENTER (FT)
C
      DIMENSION BMI(3),BPI(3),BM(3),BP(3),DISP(3)
C
C  INTERNALLY DEFINED FUNCTIONS ......
C
      TRANSM(A,B,C,D) = A+B*(C**2+D**2)
      TRANSP(A,B,C,D) = A+B*C*D
C
C  COMPUTE NEW INDIVIDUAL INERTIA PROPERTIES ......
C
      BMI(1) = TRANSM (BM(1),BMASS,DISP(2),DISP(3))
      BMI(2) = TRANSM (BM(2),BMASS,DISP(1),DISP(3))
      BMI(3) = TRANSM (BM(3),BMASS,DISP(1),DISP(2))
      BPI(1) = TRANSP (BP(1),BMASS,DISP(1),DISP(2))
      BPI(2) = TRANSP (BP(2),BMASS,DISP(1),DISP(3))
      BPI(3) = TRANSP (BP(3),BMASS,DISP(2),DISP(3))
C
      RETURN
      END
```

```
      SUBROUTINE PC (UPP,UPPD,IUPP,XPP,XPPD,IXPP,WPP,WPPD,IWPP,
     .               EPP,EPPD,IEPP,UPC,UPCD,IUPC,XPC,XPCD,IXPC,
     .               PHA,SW,FLIFT,FDRAG,FMDOT,RM,VOL,TLA,TLS,TDS,DTI,
     .               TDU,IRF,STI,RSC,RFM,RFD,RFS,B,CI,CT,CN,CM,FD,PWT,
     .               PMI,PPI,TEM,CSP,CDP,DPG,FLA,FLP,FP,TP,VAP,UVL,RL,
     .               VCG,PCG,CWT,TPE,TRM)
C
C   DESIGNED BY C.L. WEST
C   LAST MODIFIED -  DECEMBER 6, 1980
C
C   THE EASIEST PARACHUTE MODEL
C
C   ***************  PC OUTPUTS  ***************
C
C   PARACHUTE PACK LINEAR VELOCITIES - EARTH SYSTEM
C
C      UPP(3)  - X,Y,Z LINEAR VELOCITY VECTOR OF THE PARACHUTE PACK
C                CENTER OF GRAVITY (FT/SEC)
C
C      UPPD(3) - X,Y,Z LINEAR VELOCITY RATE VECTOR OF THE PARACHUTE
C                PACK CENTER OF GRAVITY (FT/SEC/SEC)
C
C      IUPP(3) - INTEGERATION CONTROL
C
C   PARACHUTE PACK LINEAR POSITIONS - EARTH SYSTEM
C
C      XPP(3)  - X,Y,Z LINEAR POSITION VECTOR OF THE PARACHUTE PACK
C                CENTER OF GRAVITY (FT)
C      XPPD(3) - X,Y,Z LINEAR POSITION RATE VECTOR OF THE PARACHUTE
C                PACK CENTER OF GRAVITY (FT/SEC)
C      IXPP(3) - INTEGRATION CONTROL
C
C   PARACHUTE PACK ANGULAR VELOCITIES - BODY AXIS
C
C      WPP(3)  - X,Y,Z ANGULAR VELOCITY COMPONENTS - P,Q,R (DEG/SEC)
C      WPPD(3) - X,Y,Z ANGULAR VELOCITY RATE COMPONENTS (DEG/SEC/SEC)
C      IWPP(3) - INTEGRATION CONTROL
C
C   EULER ANGLES  --  EARTH TO PARACHUTE PACK  --  YAW,PITCH,ROLL
C
C      EPP(3)  - EARTH TO PARACHUTE PACK EULER ANGLES (DEG)
C      EPPD(3) - EULER ANGLE RATES (DEG/SEC)
C      IEPP(3) - INTEGRATION CONTROL
C
C   PARACHUTE CANOPY LINEAR VELOCITIES - EARTH SYSTEM
C
C      UPC(3)  - X,Y,Z LINEAR VELOCITY VECTOR OF THE PARACHUTE
C                CANOPY CENTER OF GRAVITY (FT/SEC)
C      UPCD(3) - X,Y,Z LINEAR VELOCITY RATE VECTOR OF THE PARACHUTE
C                CANOPY CENTER OF GRAVITY (FT/SEC/SEC)
C      IUPC(3) - INTEGRATION CONTROL
C
C   PARACHUTE CANOPY LINEAR POSITION - EARTH SYSTEM
C
C      XPC(3)  - X,Y,Z POSITION VECTOR OF THE PARACHUTE CANOPY
C                CENTER OF GRAVITY (FT)
C      XPCD(3) - X,Y,Z POSITION RATE VECTOR OF THE PARACHUTE CANOPY
C                CENTER OF GRAVITY (FT/SEC)
```

```
C      IXPC(3) - INTEGRATION CONTROL
C
C      PHA - PARACHUTE PHASE
C                 1 = PRIOR TO PARACHUTE LAUNCH
C                 2 = FROM LAUNCH UP TO LINESTRETCH
C                 3 = AFTER LINESTRETCH
C      SW   - FLAG TO INDICATE AERODYNAMIC CALCULATION MODE
C                 0 = PRIOR TO LAUNCH
C                 1 = FROM PARACHUTE LAUNCH TO LINESTRETCH
C                 2 = DURING INFLATION
C                 3 = DURING REEFING
C                 4 = AFTER REEFING
C                 5 = PARACHUTE INFLATED
C      FLIFT(3) - X,Y,Z EARTH SYSTEM AERODYNAMIC LIFT COMPONENTS (LB)
C                     ACTING ON THE PACK BEFORE LINESTRETCH
C                     ACTING ON THE CANOPY AFTER LINESTRETCH
C      FDRAG(3) - X,Y,Z EARTH SYSTEM AEROYDYNAMIC DRAG COMPONENTS (LB)
C                     ACTING ON THE PACK BEFORE LINESTRETCH
C                     ACTING ON THE CANOPY AFTER LINESTRETCH
C      FMDOT(3) - X,Y,Z EARTH SYSTEM FORCE COMPONENTS ACTING ON THE
C                  CANOPY DUE TO AIR MASS ACQUISITION FORCE (LB)
C      RM   - RADIUS OF THE SPHERE REPRESENTING THE INFLATED CANOPY (FT)
C      VOL  - VOLUME OF THE FILLED CANOPY (FT**3)
C      TLA  - PARACHUTE LAUNCH TIME / LINE SEVERING TIME (SEC)
C      TLS  - LINESTRETCH TIME (SEC)
C      TDS  - TIME AT WHICH DISREEF OCCURS (SEC)
C      DTI  - PARACHUTE CANOPY INFLATION TIME (SEC)
C      TDU  - TIME DURATION OF REEFED PARACHUTE (SEC)
C      TRF  - THE TIME AT WHICH THE CHUTE IS REEFED (SEC)
C
C   ***************  PC INPUTS  ***************
C
C      STI      - INFLATED PARACHUTE DRAG AREA (FT**2)
C      RSC      - CIRCUMFERENCE OF THE FILLED CANOPY PLUS ONE QUARTER
C                 OF THAT DISTANCE (FT)
C      RFM      - REEF MODE FLAG
C                   0 = CHUTE IS NOT REEFED
C                   1 = TIME OF DISREEF SET AT PARACHUTE INITIATION
C                   2 = TIME OF DISREEF SET AT LINESTRETCH
C      RFD       - REEF DELAY TIME (SEC)
C      RFS       - PRODUCT OF REFERENCE AREA AND TANGENT FORCE
C                  COEFFICIENT WHEN REEFED (FT**2)
C      B         - CONSTANT USED IN THE EQUATION FOR CALCULATING
C                  SCD OF THE REEFED PARACHUTE
C      CI        - CONSTANT USED IN THE EQUATION TO COMPUTE THE CANOPY
C                  INFLATION TIME
C      CT(3)   - CONSTANTS USED IN THE EQUATION THAT CALCULATES THE
C                TANGENTIAL DRAG AREA
C      CN(3)   - CONSTANTS USED IN THE EQUATION THAT CALCULATES THE
C                NORMAL DRAG AREA
C      CM(2)   - CONSTANTS USED IN THE MACH EFFECTS EQUATION
C      FD      - WAKE TO FREE STREAM RATIO
C      PWT     - TOTAL WEIGHT OF THE PARACHUTE PACK (LB)
C      PMI(3) - PARACHUTE PACK MOMENTS OF INERTIA - IIXX,IYY,ZZ
C                (SLUGS*FT**2)
C      PPI(3) - PARACHUTE PACK PRODUCTS OF INERTIA - IXY,IXZ,IYZ
C                (SLUGS*FT**2)
C      TEM     - TIME DURATION FOR PARACHUTE EMERGENCE (SEC)
```

```
C     CSP    - PARACHUTE CANOPY SPRING CONSTANT (LB/FT)
C     CDP    - PARACHUTE CANOPY DAMPING CONSTANT (LB/FT/SEC)
C     DPG(3) - PARACHUTE PACK DAMPING AFTER MORTAR IS OFF (1/SEC)
C
C     FLA    - PARACHUTE MODE FLAG
C                    0 = PRIOR TO INITIATION
C                    1 = INITIATION
C                    2 = LAUNCH
C                    3 = MORTAR OFF
C                    4 = LINESTRETCH
C                    5 = LINES SEVERED
C     FLP(3)  - X,Y,Z FORCE COMPONENTS ACTING ON THE PARACHUTE FROM
C               THE LINES (LB)
C               (BODY AXIS FOR THE PACK  -  EARTH SYSTEM FOR THE CANOPY)
C     FP(3)   - X,Y,Z PARACHUTE PACK BODY AXIS FORCE COMPONENTS ACTING
C               ON THE PACK FROM THE MORTAR OR GUN (LB)
C     TP(3)   - X,Y,Z PARACHUTE PACK BODY AXIS TORQUE COMPONENTS ACTING
C               ON THE PACK FROM THE MORTAR OR GUN (FT-LB)
C     VAP(3) - X,Y,Z EARTH SYSTEM VELOCITY COMPONENTS OF THE
C              FORCE APPLICATION POINT (FT/SEC)
C     UVL(3) - EARTH SYSTEM PARACHUTE LINE UNIT VECTOR
C     RL     - PARACHUTE LINE LENGTH (FT)
C     VCG    - VELOCITY OF THE CANOPY CENTER OF GRAVITY ALONG THE
C              PARACHUTE LINES (FT/SEC)
C     PCG    - STRETCHED CANOPY CENTER OF GRAVITY MEASURED ALONG THE
C              PARACHUTE LINE FROM THE PARACHUTE PACK (FT)
C     CWT    - WEIGHT OF THE CANOPY DRAWN FROM THE PACK (LB)
C     TPE    - TYPE OF PARACHUTE (1=DRAG  2=RECOVERY)
C     TRM(3) - X,Y,Z PARENT BODY EARTH VELOCITY COMPONENTS
C              TO DETERMINE THE POSITION RATES DURING TRIM (FT/SEC)
C
C
C  DIMENSION OF CALLING ARGUMENTS ......
C
      DIMENSION UPP(3),UPPD(3),IUPP(3),XPP(3),XPPD(3),IXPP(3),
     .          WPP(3),WPPD(3),IWPP(3),EPP(3),EPPD(3),IEPP(3),
     .          UPC(3),UPCD(3),IUPC(3),XPC(3),XPCD(3),IXPC(3),
     .          FLP(3),FP(3),TP(3),CT(3),CN(3),CM(2),
     .          PMI(3),PPI(3),DPG(3),UVL(3),VAP(3),TRM(3)
C
C  INTERNAL DIMENSIONS ......
C
      DIMENSION EPPIR(3),WPPIR(3),FSPR(3),FDAMP(3),FPP(3),
     .          TPP(3),FPC(3),FLIFT(3),FDRAG(3),FMDOT(3),
     .          TEMP1(3),TEMP2(3),TEMP3(3),TINER(3,3),
     .          DEP(3,3),DEPT(3,3),XCG(3),UCG(3)
C
      COMMON /CICCAL/ICCAL
      COMMON /CTIME/ TIME
      COMMON /COVRLY/ INST
      COMMON /CSSFLG/ SSFLG
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
      DATA RPD,DPR /.01745329, 57.29578 /
      DATA GRAV /32.174/
C
C  ****************************
C  *****  INITIALIZATION  *****
```

```
C   ****************************
C
       IF(ICCAL.NE.1) GO TO 10
C
C   CALCULATE THE FILLED RADIUS AND VOLUME ......
C
       RM = .6366 * RSC
       VOL = 4.188 * RM**3
C
C   MISC INITIALIZATION ......
C
       DO 5 I=1,3
 5     FLIFT(I) = FDRAG(I) = FMDOT(I) = 0
       PHA = 1.
       TLA = TLS = TDS = UTI = TDU = TRF = SW = 0
       TRM(1) = TRM(2) = TRM(3) = 0
       IF(TEM.EQ.0.99999) TEM = 0
       IF(CSP.EQ.0.99999) CSP = 2000.
       IF(CDP.EQ.0.99999) CDP = 14.
       IF(RFM.EQ.0.99999) RFM = 0
       IF(RFD.EQ.0.99999) RFD = 0
       IF(RFS.EQ.0.99999) RFS = 0
       IF(RFM.EQ.0) RFD = 0
C
C   /////////////////////////////////////////////////////////////////
C
C   ----  COMPUTE THE INERTIA TENSOR  ----
C
 10    TINER(1,1) =  PMI(1)
       TINER(1,2) = -PPI(1)
       TINER(1,3) = -PPI(2)
       TINER(2,1) = -PPI(1)
       TINER(2,2) =  PMI(2)
       TINER(2,3) = -PPI(3)
       TINER(3,1) = -PPI(2)
       TINER(3,2) = -PPI(3)
       TINER(3,3) =  PMI(3)
C
C   CONVERT FROM DEGREES TO RADIANS ......
C
       DO 20 I=1,3
       EPPIR(I) = EPP(I) * RPD
 20    WPPIR(I) = WPP(I) * RPD
C
C   CALCULATE THE DIRECTION COSINE MATRICES ......
C
       CALL DIRCOS (DEP,EPPIR)
       CALL TRANS (DEPT,DEP,3,3)
C
C   DEFINE CHUTE ......
C
       IF(TPE.EQ.1.) CHUTE = 4HDRAG
       IF(TPE.EQ.2.) CHUTE = 8HRECOVERY
C
       IF(PHA.EQ.2.) GO TO 90
       IF(PHA.EQ.3.) GO TO 260
C
C   ******************************************
```

```
C   **                                          **
C   **                PHASE 1                   **
C   **                                          **
C   **      PRIOR TO PARACHUTE LAUNCH           **
C   **                                          **
C   ******************************************
C
C   ----  DEFINE VARIABLES AT PARACHUTE INITIATION AND LAUNCH  ----
C
      IF (FLA.EQ.0) GO TO 40
C
C   AT PARACHUTE INITIATION ......
C
      IF(RFM.EQ.1. .AND. TDS.EQ.0) TDS = TIME + RFD
C
C   AT PARACHUTE LAUNCH ......
C
      IF (FLA.EQ.1.0) GO TO 40
      PHA = 2.0
      SW = 1.
      TLA = TIME
      IF(INST.EQ.26) WRITE (6,30) CHUTE, TIME
 30   FORMAT(/5X,A8,* CHUTE LAUNCH AT TIME = *,F10.4,* SEC*/)
      GO TO 90
C
C   ----  DRIVE THE PARACHUTE CANOPY TO ITS CG POSITION  ----
C
C   CALCULATE THE SPRING FORCE ON THE CANOPY ......
C
 40   DO 50 I=1,3
 50   FSPR(I) = CSP * (XPP(I) - XPC(I))
C
C   CALCULATE THE DAMPING FORCE ON THE CANOPY ......
C
      DO 60 I=1,3
 60   FDAMP(I) = CDP * (UPP(I) - UPC(I))
C
C   ----  SUM FORCES AND TORQUES ACTING ON THE PARACHUTE PACK  ----
C
      DO 70 I=1,3
      FPP(I) = FP(I)
 70   TPP(I) = TP(I)
      FPP(3) = FPP(3) + PWT * SSFLG
      PMASS = PWT/GRAV
C
C   ----  SUM THE FORCES ACTING ON THE PARACHUTE CANOPY  ----
C
      DO 80 I=1,3
 80   FPC(I) = FSPR(I) + FDAMP(I)
      FPC(3) = FPC(3) + 1. * SSFLG
      CMASS = 1./GRAV
C
      GO TO 370
C
C   ************************************************
C   **                                            **
C   **                PHASE 2                     **
C   **                                            **
```

```
C  **      FROM PARACHUTE LAUNCH TO LINESTRETCH      **
C  **                                                **
C  ****************************************************
C
 90   IF (FLA.EQ.4.) GO TO 240
C
C  ----  CALCULATE THE AERODYNAMIC FORCES  ----
C
      CALL PCAERO (FLIFT,FDRAG,FMDOT,SCT,
     .             SW,XPP,UPP,TLS,DTI,TDU,VOL,UVL,CT,
     .             CN,CM,FD,B,STI,RFS,FLA,TLA,TEM)
C
C  FACTOR THE AERODYNAMIC FORCES DURING EMERGENCE ......
C
      DELTA = TIME - TLA
      IF(DELTA.GE.TEM) GO TO 120
      FACTOR = 0
      IF(TEM.NE.0) FACTOR = DELTA/TEM
      DO 110 I=1,3
      FLIFT(I) = FLIFT(I) * FACTOR
 110  FDRAG(I) = FDRAG(I) * FACTOR
C
C  ----  DRIVE THE PARACHUTE CANOPY TO ITS CG POSITION  ----
C
C  CALCULATE THE EARTH POSITION OF THE CANOPY CG ......
C
 120  DO 130 I=1,3
 130  XCG(I) = XPP(I) + PCG * UVL(I)
C
C  DETERMINE THE SPRING FORCE ACTING ON THE CANOPY ......
C
      DO 140 I=1,3
 140  FSPR(I) = CSP * (XCG(I) - XPC(I))
C
C  CALCULATE THE VELOCITY OF THE PARACHUTE PACK RELATIVE TO THE
C  FORCE APPLICATION POINT ......
C
      DO 150 I=1,3
 150  TEMP1(I) = UPP(I) - VAP(I)
C
C  DETERMINE THE VECTOR COMPONENT OF THIS RELATIVE VELOCITY NORMAL
C  TO THE LINES ......
C
      CALL DOTPRD (DIST,TEMP1,UVL,3)
      DO 160 I=1,3
 160  TEMP2(I) = DIST * UVL(I)
      DO 170 I=1,3
 170  TEMP3(I) = TEMP1(I) - TEMP2(I)
C
C  RATIO THIS VECTOR ACCORDING TO THE POSITON OF THE CANOPY CG ALONG
C  THE LINES ......
C
      RATIO = (RL-PCG)/RL
      DO 180 I=1,3
 180  TEMP3(I) = TEMP3(I) * RATIO
C
C  COMPUTE THE EARTH VELOCITY OF THE CANOPY CG POSITION ON THE
C  LINES ......
```

```
C
      DO 190 I=1,3
 190  UCG(I) = VAP(I) + TEMP3(I) - VCG * UVL(I)
C
C  DETERMINE THE EARTH VELOCITY DIFFERENCE BETWEEN THE CANOPY
C  AND THE CANOPY CG POSITION ......
C
      DO 200 I=1,3
 200  TEMP1(I) = UCG(I) - UPC(I)
C
C  CALCULATE THE DAMPING FORCE ON THE CANOPY ......
C
      DO 210 I=1,3
 210  FDAMP(I) = CDP * TEMP1(I)
C
C  ----  SUM THE FORCES AND TORQUES ACTING ON THE PARACHUTE PACK  ----
C
      WDIFF = PWT - CWT
      DO 220 I=1,3
      TPP(I) = TP(I)
 220  FPP(I) = FLIFT(I) + FDRAG(I) + FLP(I) + FP(I)
      FPP(3) = FPP(3) + WDIFF * SSFLG
      PMASS = WDIFF/GRAV
C
C  ----  SUM THE FORCES ACTING ON THE PARACHUTE CANOPY  ----
C
      DO 230 I=1,3
 230  FPC(I) = FSPR(I) + FDAMP(I)
      FPC(3) = FPC(3) + 1. * SSFLG
      CMASS = 1./GRAV
C
      GO TO 370
C
C  ----  AT LINESTRETCH  ----
C
 240  PHA = 3.
      SW = 2.
      TLS = TIME
      TLA = 0
C
C  SET DISREEF TIME ......
C
      IF(RFM.EQ.2.) TDS = TIME + RFD
C
C  CALCULATE THE CHUTE INFLATION TIME ......
C
      VBAR = SQRT(VAP(1)**2 + VAP(2)**2 + VAP(3)**2)
      DTI = CI * 2.0 * RSC/VBAR
C
C  ******************************
C  **                          **
C  **         PHASE 3          **
C  **                          **
C  **    AFTER LINESTRETCH     **
C  **                          **
C  ******************************
C
C
```

```
C ---- CALCULATE THE AERODYNAMIC FORCES ----
C
 260  GO TO (270,270,290,310,340), SW
C
C  *****  SW = 2  (DURING INFLATION)  *****
C
 270  IF(TIME.GE.TLS+DTI) GO TO 320
      GO TO 340
C
 275  IF(SCT.LT.RFS) GO TO 345
      SW = 3.
      TRF = TIME
      TDU = TDS - TIME
      IF(INST.EQ.26) WRITE(6,280) CHUTE,TIME
 280  FORMAT(/5X,A8,* CHUTE REEFED AT TIME = *,F10.4,* SEC*/)
      GO TO 345
C
C  *****  SW = 3  (DURING REEFING)  *****
C
 290  IF(TIME.LT.TDS) GO TO 340
C
      SW = 4.
      IF(INST.EQ.26) WRITE(6,300) CHUTE,TIME
 300  FORMAT(/5X,A8,* CHUTE DISREEFED AT TIME = *,F10.4,* SEC*/)
      GO TO 340
C
C  *****  SW = 4  (AFTER REEFING)  *****
C
 310  IF(TIME.GE.TLS+DTI+TDU) GO TO 320
      GO TO 340
C
C  AT THE TIME THE CANOPY IS FILLED ......
C
 320  SW = 5.
      IF(INST.EQ.26) WRITE(6,330) CHUTE,TIME
 330  FORMAT(/5X,A8,* CANOPY FILLED AT TIME = *,F10.4,* SEC*/)
C
C  DETERMINE THE LIFT AND DRAG FORCES ......
C
 340  CALL PCAERO (FLIFT,FDRAG,FMDOT,SCT,
     .                SW,XPC,UPC,TLS,DTI,TDU,VOL,UVL,CT,
     .                CN,CM,FD,B,STI,RFS,FLA,TLA,TEM)
C
      IF(RFM.NE.0 .AND. SW.EQ.2.) GO TO 275
C
C ---- SUM THE FORCES ACTING ON THE PARACHUTE CANOPY ----
C
 345  DO 350 I=1,3
 350  FPC(I) = FLIFT(I) + FDRAG(I) + FMDOT(I) + FLP(I)
      FPC(3) = FPC(3) + CWT * SSFLG
      CMASS = CWT/GRAV
C
C ---- SUM FORCES AND TORQUES ACTING ON THE PARACHUTE PACK ----
C
      WDIFF = PWT - CWT
      DO 360 I=1,3
 360  FPP(I) = TPP(I) = 0
      FPP(3) = WDIFF * SSFLG
```

```
      PMASS = WDIFF/GRAV
C
C  ************************************************
C  *****  PARACHUTE PACK EQUATIONS OF MOTION  ****
C  ************************************************
C
C  *****  PARACHUTE PACK ANGULAR VELOCITY EQUATIONS  *****
C
C  CALCULATE TINER * WPPIR
C
 370  CALL MATMPY (TEMP1,TINER,WPPIR,3,3,1)
C
C  CALCULATE WPPIR X (TINER * WPPIR)
C
      CALL CRSPRD (TEMP2,WPPIR,TEMP1)
C
C  SUM TERMS TO OBTAIN TOTAL TORQUE ......
C
      DO 380 I=1,3
 380  TEMP3(I) = TPP(I) - TEMP2(I)
C
C  CALCULATE WPPD ......
C
      CALL LUEQS (TINER,TEMP1,TEMP3,TEMP2,3,1,3,3,3,1.E-14,IERROR)
      IF(IERROR.NE.1) GO TO 400
      WRITE(6,390) CHUTE
 390  FORMAT(* INERTIA MATRIX OF *A8* CHUTE IS SINGULAR...RUN STOPPED*)
      STOP
C
 400  DO 410 I=1,3
      IF(IWPP(I).NE.0) WPPD(I) = TEMP1(I) * DPR
 410  IF(FLA.GT.3.0.AND.IWPP(I).NE.0) WPPD(I) = -DPG(I) * WPP(I)
C
C  *****  PARACHUTE PACK EULER ANGLE EQUATIONS  *****
C
      CALL EARATE (TEMP1,WPPIR,EPPIR)
      DO 420 I=1,3
 420  IF(IEPP(I).NE.0) EPPD(I) = TEMP1(I) * DPR
C
C  *****  PARACHUTE PACK LINEAR VELOCITY EQUATIONS  *****
C
      DO 430 I=1,3
 430  IF(IUPP(I).NE.0) UPPD(I) = FPP(I)/PMASS
C
C  *****  PARACHUTE PACK LINEAR POSITION EQUATIONS  *****
C
      DO 450 I=1,3
 450  IF(IXPP(I).NE.0) XPPD(I) = UPP(I)
C
C  DURING TRIM, SUBTRACT TRIM VELOCITY FROM POSITION RATES
C
      IF(INST.NE.31) GO TO 470
      DO 460 I=1,3
 460  IF(IXPP(I).NE.0) XPPD(I) = XPPD(I) - TRM(I)
 470  CONTINUE
C
C  ************************************************
C  *****  PARACHUTE CANOPY EQUATIONS OF MOTION  *****
```

```
C   *************************************************
C
C   ----  LINEAR VELOCITY EQUATIONS  ----
C
      DO 480 I=1,3
 480  IF(IUPC(I).NE.0) UPCD(I) = FPC(I)/CMASS
C
C   ----  LINEAR POITION EQUATIONS  ----
C
      DO 490  I=1,3
 490  IF(IXPC(I).NE.0) XPCD(I) = UPC(I)
C
C   ----  DURING TRIM SUBTRACT TRIM VELOCITY FROM POSITION RATES  ----
C
      IF(INST.NE.31) GO TO 510
      DO 500 I=1,3
 500  IF(IXPC(I).NE.0) XPCD(I) = XPCD(I) - TRM(I)
 510  CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE RL (FRS,TRS,FRA,TRA,FL,FTS,TTS,OFF,DSA,SRA,DIS,TM,
     .               BL1,BL2,BL3,BL4,BL5,BL6,
     .               UP,RLR,XRR,RLL,XRL,ERL,SPR,DPG,SBF,ZTS,BTS,CPT,
     .               SRP,UST,EST,WST,XAP,UAP,EAP,WAP)
C
      COMMON /CTIME/ TIME
      COMMON /CICCAL/ ICCAL
      COMMON /COVRLY/ INST
      COMMON /CSSFLG/ SSFLG
      COMMON / CIO / IREAD,IWRITE,IDIAG
C
C  DESIGNED BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C  FORCES AND TORQUES ON THE VEHICLE AND SEAT FROM RAIL ELASTICITY AND
C  RAIL TO SLIDER BLOCK FRICTION FORCES
C  BLOCKS STARTING AT THE BOTTOM OF THE RIGHT RAIL AND GOING UP ARE
C  NUMBERED 1, 2, 3; AT THE BOTTOM OF THE LEFT RAIL AND GOING UP ARE
C  NUMBERED 4, 5, 6
C
C  ***************  RAIL OUTPUTS  ***************
C
C     FRS(3)    - X,Y,Z SEAT AXIS FORCE COMPONENTS ON THE SEAT FROM
C                 THE RAILS (LB)
C     TRS(3)    - X,Y,Z SEAT AXIS TORQUE COMPONENTS ON THE SEAT FROM
C                 THE RAILS (FT-LB)
C     FRA(3)    - X,Y,Z AIRPLANE AXIS FORCE COMPONENTS ON THE AIRPLANE
C                 FROM THE RAILS (LB)
C     TRA(3)    - X,Y,Z AIRPLANE AXIS TORQUE COMPONENTS ON THE AIRPLANE
C                 FROM THE RAILS (FT-LB)
C     FL        - STROKE FLAG (0 = GUIDED     1 = UNGUIDED)
C     FTS       - TRIP SWITCH CONTACT FLAG (1 = ON)
C     TTS       - TRIP SWITCH CONTACT TIME (SEC)
C     OFF       - SEAT/RAIL SEPARATION FLAG (1 = SEPARATION)
C     DSA(3,3)  - SEAT TO AIRPLANE DIRECTION COSINE MATRIX
C     SRA(3)    - X,Y,Z AIRPLANE COORDINATE SYSTEM LINEAR POSITION
C                 VECTOR OF THE SRP (FT)
C     DIS       - DISTANCE FROM THE CRITICAL POINT TO THE SEAT
C                 REFERENCE POINT (FT)
C     TM(3)     - X,Y,Z VEHICLE EARTH VELOCITY COMPONENTS TO PASS
C                 TO THE SEAT COMPONENT DURING TRIM (FT/SEC)
C
C  ***************  RAIL INPUTS  ***************
C
C     BL1(3) - X,Y,Z SEAT AXIS POSITION VECTOR OF RIGHT LOWER BLOCK (FT)
C     BL2(3) - X,Y,Z SEAT AXIS POSITION VECTOR OF RIGHT MIDDLE BLOCK (FT)
C     BL3(3) - X,Y,Z SEAT AXIS POSITION VECTOR OF RIGHT UPPER BLOCK (FT)
C     BL4(3) - X,Y,Z SEAT AXIS POSITION VECTOR OF LEFT LOWER BLOCK (FT)
C     BL5(3) - X,Y,Z SEAT AXIS POSITION VECTOR OF LEFT MIDDLE BLOCK (FT)
C     BL6(3) - X,Y,Z SEAT AXIS POSITION VECTOR OF LEFT UPPER BLOCK (FT)
C
C     UP     - EJECTION DIRECTION FLAG
C                   +1 = UPWARD WRT THE VEHICLE
C                   -1 = DOWNWARD WRT THE VEHICLE
C     RLR    - RIGHT RAIL Z COORDINATE OF THE END OF THE RIGHT RAIL (FT)
C     XRR(3) - X,Y,Z AIRPLANE POSITION VECTOR OF THE ORIGIN OF
C              THE RIGHT RAIL COORDINATE SYSTEM (FT)
C     RLL    - LEFT RAIL Z COORDINATE OF THE END OF THE LEFT RAIL (FT)
```

```
C      XRL(3) - X,Y,Z AIRPLANE POSITION VECTOR OF THE ORIGIN OF
C               THE LEFT RAIL COORDINATE SYSTEM (FT)
C      ERL(3) - AIRPLANE TO RAILS EULER ANGLES (DEG)
C      SPR(2) - RAIL SPRING CONSTANT (LB/FT)
C      DPG(2) - RAIL DAMPING CONSTANT (LB/FT/SEC)
C      SBF    - SLIDER BLOCK FRICTION COEFFICIENT
C      ZTS    - RIGHT RAIL AXIS Z COORDINATE OF THE KEY BLOCK AT
C               TRIP SWITCH CONTACT (FT)
C      BTS    - TRIP SWITCH KEY BLOCK NUMBER
C                         1 = BOTTOM RIGHT BLOCK
C                         2 = MIDDLE RIGHT BLOCK
C                         3 = TOP RIGHT BLOCK
C      CPT(3) - X,Y,Z AIRPLANE POSITION VECTOR OF THE CRITICAL CLEARANCE
C               POINT FOR THE SEAT (FT)
C
C      SRP(3) - X,Y,Z EARTH POSITION VECTOR OF THE SEAT REFERENCE POINT (FT)
C      UST(3) - X,Y,Z SEAT VELOCITY VECTOR OF THE SRP (FT/SEC)
C      EST(3) - EARTH TO SEAT EULER ANGLES (DEG)
C      WST(3) - X,Y,Z SEAT ANGULAR VELOCITY FECTOR OF THE SEAT (DEG/SEC)
C      XAP(3) - X,Y,Z EARTH POSITION VECTOR OF THE AIRPLANE (FT)
C      UAP(3) - X,Y,Z AIRPLANE VELOCITY VECTOR OF THE AIRPLANE (FT/SEC)
C      EAP(3) - EARTH TO AIRPLANE EULER ANGLES (DEG)
C      WAP(3) - X,Y,Z AIRPLANE ANGULAR VELOCITY VECTOR OF THE AIRPLANE
C               (DEG/SEC)
C
C   DIMENSIONS OF CALLING ARGUMENTS ......
C
       DIMENSION FRS(3),TRS(3),FRA(3),TRA(3),DSA(3,3),SRA(3),TM(3),
     .            BL1(3),BL2(3),BL3(3),BL4(3),BL5(3),BL6(3),XRR(3),
     .            XRL(3),ERL(3),SPR(2),DPG(2),CPT(3),SRP(3),UST(3),EST(3),
     .            WST(3),XAP(3),UAP(3),EAP(3),WAP(3)
C
C   INTERNAL DIMENSIONS ......
C
       DIMENSION DAR(3,3),DRA(3,3),DEA(3,3),DAE(3,3),DES(3,3),
     .            DSE(3,3),DER(3,3),DRE(3,3),DRS(3,3),DSR(3,3),
     .            ESTIR(3),EAPIR(3),WSTIR(3),WAPIR(3),ERLIR(3),
     .            SRPRR(3),SRPRL(3),SW(6)
C
       DIMENSION FSB1(3),FAB1(3),TSB1(3),TAB1(3),
     .            FSB2(3),FAB2(3),TSB2(3),TAB2(3),
     .            FSB3(3),FAB3(3),TSB3(3),TAB3(3),
     .            FSB4(3),FAB4(3),TSB4(3),TAB4(3),
     .            FSB5(3),FAB5(3),TSB5(3),TAB5(3),
     .            FSB6(3),FAB6(3),TSB6(3),TAB6(3)
C
       DATA RPD / .01745329 /
C
C   ****************************
C   *****  INITIALIZATION  *****
C   ****************************
C
       IF(ICCAL.NE.1) GO TO 5
C
C   INITIALIZE VARIABLES ......
C
       OFF = FL = FTS = TTS = 0
       TM(1) = TM(2) = TM(3) = 0
```

```
      IF(CPT(1).EQ.0.99999) CPT(1) = 0
      IF(CPT(2).EQ.0.99999) CPT(2) = 0
      IF(CPT(3).EQ.0.99999) CPT(3) = 0
      IF(UP.EQ.0.99999) UP = 1.
      IF(BTS.EQ.0.99999) BTS = 1.
C
C  *****************************************************
C
 5    DO 10 I=1,3
 10   EAPIR(I) = EAP(I) * RPD
      CALL DIRCOS (DEA,EAPIR)
C
C  CALCULATE SEAT REFERENCE POINT COORDINATES IN THE AIRPLANE SYSTEM ......
C
      CALL VECXYZ (SRA,SRP,XAP,DEA,1)
C
C  CALCULATE THE DISTANCE FROM THE CRITICAL POINT TO THE SRP ......
C
      DIS = SQRT((CPT(1)-SRA(1))**2 + (CPT(2)-SRA(2))**2 +
     .       (CPT(3)-SRA(3))**2)
C
C  RETURN TO EQMO IF SEAT BLOCKS ARE OFF RAILS
C
      IF(OFF.EQ.1.0) GO TO 140
C
C  CHANGE FROM DEGREES TO RADIANS ......
C
      DO 20 I=1,3
      ESTIR(I) = EST(I) * RPD
      WSTIR(I) = WST(I) * RPD
      WAPIR(I) = WAP(I) * RPD
 20   ERLIR(I) = ERL(I) * RPD
C
C  CALCULATE THE DIRECTION COSINE MATRICES ......
C
      CALL DIRCOS (DAR,ERLIR)
      CALL TRANS (DRA,DAR,3,3)
      CALL TRANS (DAE,DEA,3,3)
      CALL DIRCOS (DES,ESTIR)
      CALL TRANS (DSE,DES,3,3)
      CALL MATMPY (DER,DAR,DEA,3,3,3)
      CALL TRANS (DRE,DER,3,3)
      CALL MATMPY (DRS,DES,DRE,3,3,3)
      CALL TRANS (DSR,DRS,3,3)
      CALL MATMPY (DSA,DEA,DSE,3,3,3)
C
C  ************************************************************
C  ****  SLIDER BLOCK FORCES AND TORQUES FOR THE RIGHT RAIL  ****
C  ************************************************************
C
C
C  DETERMINE SEAT REFERENCE POINT IN RIGHT RAIL SYSTEM ......
C
      CALL VECXYZ (SRPRR,SRA,XRR,DAR,1)
C
C  BOTTOM BLOCK (1)
C
      CALL BLOCK (FSB1,FAB1,TSB1,TAB1,ZB1,SRPRR,SRA,RLR,XRR,BL1,SPR,
```

```
     .              DPG,SBF,UST,WSTIR,UAP,WAPIR,DAE,DER,DRS,DRA,DSA,DSE,
     .              DSR,UP,SW(1))
      IF(INST.NE.26 .OR. BTS.NE.1.) GO TO 30
      IF(ZTS*UP .GE. ZB1*UP) FTS = 1.0
      IF(FTS.EQ.1.0 .AND. TTS.EQ.0) TTS = TIME
C
C  MIDDLE BLOCK ( 2)
C
 30   CALL BLOCK (FSB2,FAB2,TSB2,TAB2,ZB2,SRPRR,SRA,RLR,XRR,BL2,SPR,
     .              DPG,SBF,UST,WSTIR,UAP,WAPIR,DAE,DER,DRS,DRA,DSA,DSE,
     .              DSR,UP,SW(2))
      IF(INST.NE.26 .OR. BTS.NE.2.) GO TO 40
      IF(ZTS*UP .GE. ZB2*UP) FTS = 1.0
      IF(FTS.EQ.1.0 .AND. TTS.EQ.0) TTS = TIME
C
C  TOP BLOCK ( 3)
C
 40   CALL BLOCK (FSB3,FAB3,TSB3,TAB3,ZB3,SRPRR,SRA,RLR,XRR,BL3,SPR,
     .              DPG,SBF,UST,WSTIR,UAP,WAPIR,DAE,DER,DRS,DRA,DSA,DSE,
     .              DSR,UP,SW(3))
      IF(INST.NE.26 .OR. BTS.NE.3.) GO TO 50
      IF(ZTS*UP .GE. ZB3*UP) FTS = 1.0
      IF(FTS.EQ.1.0 .AND. TTS.EQ.0) TTS = TIME
C
C  ***************************************************************
C  ****  SLIDER BLOCK FORCES AND TORQUES FOR THE LEFT RAIL  ****
C  ***************************************************************
C
C  DETERMINE SEAT REFERENCE POINT IN THE LEFT RAIL SYSTEM
C
 50   CALL VECXYZ (SRPRL,SRA,XRL,DAR,1)
C
C  BOTTOM BLOCK ( 4)
C
      CALL BLOCK (FSB4,FAB4,TSB4,TAB4,DUM,SRPRL,SRA,RLL,XRL,BL4,SPR,
     .              DPG,SBF,UST,WSTIR,UAP,WAPIR,DAE,DER,DRS,DRA,DSA,DSE,
     .              DSR,UP,SW(4))
C
C  MIDDLE BLOCK ( 5)
C
      CALL BLOCK (FSB5,FAB5,TSB5,TAB5,DUM,SRPRL,SRA,RLL,XRL,BL5,SPR,
     .              DPG,SBF,UST,WSTIR,UAP,WAPIR,DAE,DER,DRS,DRA,DSA,DSE,
     .              DSR,UP,SW(5))
C
C  UPPER BLOCK ( 6)
C
      CALL BLOCK (FSB6,FAB6,TSB6,TAB6,DUM,SRPRL,SRA,RLL,XRL,BL6,SPR,
     .              DPG,SBF,UST,WSTIR,UAP,WAPIR,DAE,DER,DRS,DRA,DSA,DSE,
     .              DSR,UP,SW(6))
C
C  ****************************************
C  ****  CHECK IF BLOCKS ARE OFF RAILS  ****
C  ****************************************
C
      IF(FL.EQ.1.) GO TO 70
      IF(SW(2).NE.1. .AND. SW(5).NE.1.) GO TO 70
C
C  WRITE END OF GUIDED STROKE ON OUTPUT FILE
```

```
C
      IF(INST.EQ.26) WRITE(6,60) TIME
 60   FORMAT(/5X,*END OF GUIDED STROKE AT TIME = *,F7.4,* SEC*/)
      FL = 1.
C
C  ***********************************************
C  ****  TOTAL FORCES AND MOMENTS ON THE SEAT  ****
C  ***********************************************
C
 70   DO 80 I=1,3
      FRS(I) = FSB1(I)+FSB2(I)+FSB3(I)+FSB4(I)+FSB5(I)+FSB6(I)
      TRS(I) = TSB1(I)+TSB2(I)+TSB3(I)+TSB4(I)+TSB5(I)+TSB6(I)
 80   CONTINUE
C
C  TOTAL FORCES AND MOMENTS ON AIRPLANE
C
      DO 90 I=1,3
      FRA(I) = (FAB1(I)+FAB2(I)+FAB3(I)+FAB4(I)+FAB5(I)+FAB6(I))
     .           * SSFLG
      TRA(I) = (TAB1(I)+TAB2(I)+TAB3(I)+TAB4(I)+TAB5(I)+TAB6(I))
     .           * SSFLG
 90   CONTINUE
C
C  IF FOUR OUTER BLOCKS ARE OFF RAILS, SET FLAG TO BYPASS THIS
C  COMPONENT ....
C
      IF(SW(1)+SW(3)+SW(4)+SW(6).EQ.4) OFF=1.0
      IF(OFF.EQ.0) GO TO 130
C
C  WRITE SEAT/RAIL SEPARATION MESSAGE ......
C
      IF(INST.EQ.26) WRITE(6,100) TIME
 100  FORMAT(/5X,*SEAT/RAIL SEPARATION AT TIME = *,F7.4,* SEC*/)
      DO 110 I=1,3
      FRS(I)=0.
      TRS(I)=0.
      FRA(I)=0.
      TRA(I)=0.
      SRA(I) = 0.
 110  CONTINUE
      DO 120 I=1,3
      DO 120 J=1,3
 120  DSA(I,J) = 0
C
C  SEND DATA TO DETERMINE TRIM EARTH VELOCITY TO SEAT ......
C
 130  IF(INST.NE.31)GO TO 140
      CALL MATMPY (TM,DAE,DAP,3,3,1)
C
 140  RETURN
      END
```

```
      SUBROUTINE RS (FPB,TPB,FAB,TAB,TRM,
     .                 FL,XYZ,EA,XPB,UPB,EPB,WPB,XAB,UAB,EAB,WAB,
     .                 XR,XD,ER,ED)
      COMMON / CICCAL / ICCAL
      COMMON / COVRLY / INST
      COMMON / CSSFLG / SSFLG
      COMMON / CIO / IREAD,IWRITE,IDIAG
C
C  STANDARD COMPONENT RS GENERATES THE FORCES AND TORQUES THAT
C  RESTRAINS ONE BODY TO ANOTHER (THE MAN IN THE SEAT, ETC.)
C
C  ***************  RS OUTPUTS  ***************
C
C      FPB(3) - X,Y,Z PARENT BODY AXIS FORCE VECTOR (LB)
C      TPB(3) - X,Y,Z PARENT BODY AXIS TORQUE VECTOR (FT-LB)
C      FAB(3) - X,Y,Z ATTACHED BODY AXIS FORCE VECTOR (LB)
C      TAB(3) - X,Y,Z ATTACHED BODY AXIS TORQUE VECTOR (FT-LB)
C      TRM(3) - X,Y,Z PARENT BODY EARTH VELOCITY COMPONENTS
C               TO PASS TO THE ATTACHED BODY DURING TRIM (FT/SEC)
C
C  ***************  RS INPUTS  ***************
C
C      FL     - FLAG TO RELEASE ATTACHED BODY (1 = RELEASE)
C      XYZ(3) - X,Y,Z BODY AXIS POSITON VECTOR OF THE ATTACHED
C               BODY IN THE PARENT SYSTEM (FT)
C      EA(3)  - PARENT BODY TO ATTACHED BODY EULER ANGLES (DEG)
C      XPB(3) - X,Y,Z EARTH SYSTEM POSITON VECTOR OF THE PARENT
C               BODY (FT)
C      UPB(3) - X,Y,Z PARENT BODY AXIS VELOCITY VECTOR OF THE
C               PARENT BODY (FT/SEC)
C      EPB(3) - EARTH TO PARENT BODY EULER ANGLES (DEG)
C      WPB(3) - X,Y,Z BODY AXIS ANGULAR VELOCITY VECTOR OF
C               THE PARENT BODY (DEG/SEC)
C      XAB(3) - X,Y,Z EARTH SYSTEM POSITON VECTOR OF THE ATTACHED
C               BODY (FT)
C      UAB(3) - X,Y,Z BODY AXIS VELOCITY VECTOR OF THE ATTACHED
C               BODY (FT/SEC)
C      EAB(3) - EARTH TO ATTACHED BODY EULER ANGLES (DEG)
C      WAB(3) - X,Y,Z BODY AXIS ANGULAR VELOCITY VECTOR OF
C               THE ATTACHED BODY (DEG/SEC)
C      XR     - LINEAR SPRING CONSTANT (LB/FT)
C      XD     - LINEAR DAMPING CONSTANT (LB/FT/SEC)
C      ER(3)  - X,Y,Z,ANGULAR SPRING CONSTANTS (FT-LB/DEG)
C      ED(3)  - X,Y,Z ANGULAR DAMPING CONSTANTS (FT-LB/DEG/SEC)
C
C  DIMENSIONS OF CALLING ARGUMENTS ......
C
      DIMENSION FPB(3),TPB(3),FAB(3),TAB(3),TRM(3),
     .          XYZ(3),EA(3),XPB(3),UPB(3),EPB(3),WPB(3),
     .          XAB(3),UAB(3),EAB(3),WAB(3),ER(3),ED(3)
C
C  INTERNAL DIMENSIONS ......
C
      DIMENSION DPB(3,3),DAB(3,3),DPBT(3,3),DPA(3,3),
     .          XB(3),DELTA(3),SPRING(3),UXE(3),VEL(3),
     .          DAMP(3),ANG(3),TORQUE(3),WPBE(3),WABE(3),
     .          EPBIR(3),WPBIR(3),EABIR(3),WABIR(3),DABE(3,3),UABE(3),
     .          EAIR(3),DCEA(3,3),DCEAT(3,3),TEMP(3)
```

```
C
      DATA RPD,DPR / .01745329, 57.29578 /
C
C   ***************************
C   *****  INITIALIZATION  *****
C   ***************************
C
      IF(ICCAL.NE.1) GO TO 5
      TRM(1) = TRM(2) = TRM(3) = 0
      ISW = 0
      DO 2 I=1,3
 2    EAIR(I) = EA(I) * RPD
      CALL DIRCOS (DCEA,EAIR)
      CALL TRANS (DCEAT,DCEA,3,3)
C
C   BYPASS CALCULATIONS IF THE FLAG IS SET TO RELEASE ......
C
 5    IF(ISW.EQ.1) GO TO 140
      IF(FL.NE.1) GO TO 20
      DO 10 I=1,3
      FPB(I) = 0.
      FAB(I) = 0.
      TPB(I) = 0.
 10   TAB(I) = 0.
      ISW = 1
      GO TO 140
C
C   *****  CHANGE FROM DEGREES TO RADIANS  *****
C
 20   DO 30 I=1,3
      EPBIR(I) = EPB(I) * RPD
      WPBIR(I) = WPB(I) * RPD
      EABIR(I) = EAB(I) * RPD
 30   WABIR(I) = WAB(I) * RPD
C
C   CALCULATE THE DIRECTION COSINE MATRICES ......
C
      CALL DIRCOS (DPB,EPBIR)
      CALL DIRCOS (DAB,EABIR)
      CALL TRANS (DABE,DAB,3,3)
      CALL TRANS (DPBT,DPB,3,3)
      CALL MATMPY (DPA,DAB,DPBT,3,3,3)
C
C   *******************************************************
C   *****  FORCES AND TORQUES DUE TO LINEAR DISPLACEMENT  *****
C   *******************************************************
C
C   CALCULATE THE ATTACHED BODY LINEAR POSITION VECTOR IN THE
C   PARENT BODY COORDINATE SYSTEM (XB) .....
C
      CALL VECXYZ (XB,XAB,XPB,DPB,1)
C
C   DETERMINE THE LINEAR DISPLACEMENT FROM THE ATTACHMENT POINT,
C   AND CALCULATE THE BODY AXIS SPRING COMPONENTS ACTING ON THE
C   PARENT BODY ......
C
      DO 40 I=1,3
      DELTA(I) = XB(I) - XYZ(I)
```

```
 40    SPRING(I) = DELTA(I) * XR
C
C  CALCULATE THE BODY AXIS DAMPING COMPONENTS ACTING ON THE PARENT
C  BODY, AND SUM THE RESULTS WITH THE SPRING COMPONENTS ......
C
       CALL VELXYZ (UXE,UPB,XB,WPBIR,DPBT)
       CALL MATMPY (UABE,DABE,UAB,3,3,1)
       DO 50 I=1,3
 50    DELTA(I) = UABE(I) - UXE(I)
       CALL MATMPY (VEL,DPB,DELTA,3,3,1)
       DO 60 I=1,3
       DAMP(I) = VEL(I) * XD
 60    FPB(I) = SPRING(I) + DAMP(I)
C
C  CALC TORQUE ON PARENT BODY DUE TO DISPLACEMENT OF ATTACHMENT
C  POINT FROM PARENT BODY CENTER OF GRAVITY
C
       CALL CRSPRD (TORQUE,XYZ,FPB)
C
C  CALCULATE THE BODY AXIS FORCE COMPONENTS ACTING ON THE
C  ATTACHED BODY ......
C
       CALL MATMPY (FAB,DPA,FPB,3,3,1)
       DO 70 I=1,3
 70    FAB(I) = -FAB(I)
C
C  ************************************************
C  *****  TORQUE DUE TO ANGULAR DISPLACEMENT  *****
C  ************************************************
C
C  CALCULATE THE PARENT TO ATTACHED BODY EULER ANGLES ......
C
       CALL COSDIR (ANG,DPA)
C
C  DETERMINE THE ANGULAR DISPLACEMENT FROM THE ATTACHMENT ANGLE,
C  AND CALCULATE THE SPRING COMPONENTS ACTING ON THE SEAT IN THE
C  ATTACHMENT AXIS SYSTEM ......
C
       DO 80 I=1,3
       DELTA(I) = ANG(4-I)*DPR - EA(4-I)
 80    SPRING(I) = DELTA(I) * ER(I)
C
C  CALCULATE THE BODY AXIS ANGULAR DAMPING COMPONENTS ACTING
C  ON THE PARENT BODY IN THE ATTACHMENT AXIS SYSTEM ......
C
       CALL MATMPY (WPBE,DPBT,WPB,3,3,1)
       CALL MATMPY (WABE,DABE,WAB,3,3,1)
       DO 90 I=1,3
 90    DELTA(I) = WABE(I) - WPBE(I)
       CALL MATMPY (TEMP,DPB,DELTA,3,3,1)
       CALL MATMPY (VEL,DCEA,TEMP,3,3,1)
       DO 100 I=1,3
       DAMP(I) = VEL(I) * ED(I)
 100   TEMP(I) = SPRING(I) + DAMP(I)
C
C  MOVE THE RESTRAINT TORQUES INTO THE SEAT SYSTEM ......
C
       CALL MATMPY (TPB,DCEAT,TEMP,3,3,1)
```

```
C
C  CALCULATE THE BODY AXIS TORQUE COMPONENTS ACTING ON THE
C  ATTACHED BODY ......
C
      CALL MATMPY (TAB,DPA,TPB,3,3,1)
      DO 110 I=1,3
 110  TAB(I) = -TAB(I)
C
C  CALCULATE THE TOTAL MOMENT ON THE PARENT BODY ......
C
      DO 120 I=1,3
 120  TPB(I) = TPB(I) + TORQUE(I)
C
C  ZERO THE FORCES AND TORQUES ACTING ON THE PARENT BODY IF SSFLG IS
C  EQUAL TO ZERO ......
C
      IF(SSFLG.NE.0) GO TO 135
      DO 130 I=1,3
 130  FPB(I) = TPB(I) = 0
C
C  SEND DATA TO ATTACHED BODY TO ALLOW IT TO COMPUTE THE PARENT BODY
C  EARTH VELOCITY DURING TRIM ......
C
 135  IF (INST.NE.31) GO TO 140
      CALL MATMPY (TRM,DPBT,UPB,3,3,1)
C
 140  RETURN
      END
```

```
      SUBROUTINE SE (UST,UDS,IUS,SRP,XDS,IXS,WST,WDS,IWS,
     .              EST,EDS,IES,SCD,SCDDOT,ISCD,SC,SCDOT,ISC,
     .              GX,GY,GZ,DR,ALT,
     .              F11,F12,F13,F14,F15,F16,F17,F18,
     .              T11,T12,T13,T14,T15,T16,T17,T18,
     .              F21,F22,F23,F24,F25,F26,F27,F28,
     .              T21,T22,T23,T24,T25,T26,T27,T28,
     .              CW,CCG,CMI,CPI,TM)
C
C
C        EASIEST SEAT EQUATIONS OF MOTION COMPONENT
C
C
C   DESIGNED BY C.L. WEST
C   LAST MODIFIED -  DECEMBER 6, 1980
C
C   ***************  SE OUTPUTS  ***************
C
C   LINEAR VELOCITIES - BODY AXIS
C
C      UST(3) - X,Y,Z  LINEAR VELOCITY VECTOR OF THE SEAT
C               REFERENCE POINT (FT/SEC)
C      UDS(3) - X,Y,Z LINEAR ACCELERATION VECTOR OF THE SEAT
C               REFERENCE POINT (FT/SEC/SEC)
C      IUS(3) - INTEGRATION CONTROL
C
C   LINEAR POSITIONS - EARTH SYSTEM
C
C      SRP(3) - X,Y,Z LINEAR POSITION VECTOR OF THE SEAT
C               REFERENCE POINT (FT)
C      XDS(3) - X,Y,Z LINEAR VELOCITY VECTOR OF THE SEAT
C               REFERENCE POINT (FT/SEC)
C      IXS(3) - INTEGRATION CONTROL
C
C   ANGULAR VELOCITIES - BODY AXIS
C
C      WST(3) - X,Y,Z ANGULAR VELOCITY COMPONENTS - P,Q,R  (DEG/SEC)
C      WDS(3) - X,Y,Z ANGULAR ACCELERATION COMPONENTS (DEG/SEC/SEC)
C      IWS(3) - INTEGRATION CONTROL
C
C   EULER ANGLES  --  EARTH TO BODY  --  YAW,PITCH,ROLL
C
C      EST(3) - EARTH TO SEAT EULER ANGLES (DEG)
C      EDS(3) - EULER ANGLE RATES (DEG/SEC)
C      IES(3) - INTEGRATION CONTROL
C
C   SPINAL COMPRESSION VELOCITY ......
C
C      SCD    - SPINAL COMPRESSION VELOCITY (FT/SEC)
C      SCDDOT - SPINAL COMPRESSION VELOCITY RATE (FT/SEC/SEC)
C      ISCD   - INTEGRATION CONTROL
C
C   SPINAL COMPRESSION ......
C
C      SC     - SPINAL COMPRESSION (FT)
C      SCDOT  - SPINAL COMPRESSION RATE (FT/SEC)
C      ISC    - INTEGRATION CONTROL
C
```

```
C     GX      - SEAT X-AXIS LOAD FACTOR (G)
C     GY      - SEAT Y-AXIS LOAD FACTOR (G)
C     GZ      - SEAT Z-AXIS LOAD FACTOR (G)
C     DR      - DYNAMIC RESPONSE
C     ALT     - SEAT ALTITUDE (FT)
C
C  ***************  SC INPUTS  ***************
C
C     F11(3) THROUGH F18(3) - SEAT AXIS FORCE VECTORS ACTING ON THE
C                             EJECTION SEAT WHICH ARE GENERATED BY
C                             AN EXPLOSIVE CHARGE (LB)
C     T11(3) THROUGH T18(3) - SEAT AXIS TORQUE VECTORS ACTING ON THE
C                             EJECTION SEAT WHICH ARE GENERATED BY
C                             AN EXPLOSIVE CHARGE (FT-LB)
C     F21(3) THROUGH F28(3) - SEAT AXIS FORCE VECTORS ACTING ON THE
C                             EJECTION SEAT WHICH ARE GENERATED BY
C                             NON-EXPLOSIVE MEANS (LB)
C     T21(3) THROUGH T28(3) - SEAT AXIS TORQUE VECTORS ACTING ON THE
C                             EJECTION SEAT WHICH ARE GENERATED BY
C                             NON-EXPLOSIVE MEANS (FT-LB)
C     CW      - COMPOSITE SEAT WEIGHT (LB)
C     CCG(3) - X,Y,Z SEAT AXIS SYSTEM COMPOSITE CENTER OF GRAVITY (FT)
C     CMI(3) - COMPOSITE SEAT MOMENT OF INERTIA VECTOR ABOUT ITS
C              CENTER OF GRAVITY - IXX,IYY,IZZ (SLUG-FT**2)
C     CPI(3) - COMPOSITE SEAT PRODUCT OF INERTIA VECTOR ABOUT ITS
C              CENTER OF GRAVITY - IXY,IXZ,IYZ (SLUG-FT**2)
C     TM(3)   - X,Y,Z VEHICLE EARTH VELOCITY COMPONENTS TO
C              DETERMINE THE POSITION RATES DURING TRIM (FT/SEC)
C
C  DIMENSIONS OF CALLING ARGUMENTS ......
C
      DIMENSION UST(3),UDS(3),IUS(3),SRP(3),XDS(3),IXS(3),
     .          WST(3),WDS(3),IWS(3),EST(3),EDS(3),IES(3),
     .          CCG(3),CMI(3),CPI(3),TM(3)
      DIMENSION F11(3),F12(3),F13(3),F14(3),F15(3),F16(3),F17(3),F18(3),
     .          T11(3),T12(3),T13(3),T14(3),T15(3),T16(3),T17(3),T18(3)
C
      DIMENSION F21(3),F22(3),F23(3),F24(3),F25(3),F26(3),F27(3),F28(3),
     .          T21(3),T22(3),T23(3),T24(3),T25(3),T26(3),T27(3),T28(3)
C
C  INTERNAL DIMENSIONS ......
C
      DIMENSION TINER(3,3),TEMP1(3),TEMP2(3),TEMP3(3),TEMP4(3),DSE(3,3),
     .         F(3),T(3),WSTIR(3),WDSIR(3),ESTIR(3),DES(3,3),FG(3),TG(3)
C
      COMMON /CICCAL/ ICCAL
      COMMON /COVRLY/ INST
      COMMON /CSSFLG/ SSFLG
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
      DATA RPD,DPR / .01745329, 57.29578 /
      DATA GRAV /32.174/
C
C  ****************************
C  *****  INITIALIZATION  *****
C  ****************************
C
      IF(ICCAL.NE.1) GO TO 20
```

```
C
      DO 10 I=1,3
      IF(F11(I) .EQ. 0.99999) F11(I) = 0
      IF(F12(I) .EQ. 0.99999) F12(I) = 0
      IF(F13(I) .EQ. 0.99999) F13(I) = 0
      IF(F14(I) .EQ. 0.99999) F14(I) = 0
      IF(F15(I) .EQ. 0.99999) F15(I) = 0
      IF(F16(I) .EQ. 0.99999) F16(I) = 0
      IF(F17(I) .EQ. 0.99999) F17(I) = 0
      IF(F18(I) .EQ. 0.99999) F18(I) = 0
      IF(F21(I) .EQ. 0.99999) F21(I) = 0
      IF(F22(I) .EQ. 0.99999) F22(I) = 0
      IF(F23(I) .EQ. 0.99999) F23(I) = 0
      IF(F24(I) .EQ. 0.99999) F24(I) = 0
      IF(F25(I) .EQ. 0.99999) F25(I) = 0
      IF(F26(I) .EQ. 0.99999) F26(I) = 0
      IF(F27(I) .EQ. 0.99999) F27(I) = 0
      IF(F28(I) .EQ. 0.99999) F28(I) = 0
      IF(T11(I) .EQ. 0.99999) T11(I) = 0
      IF(T12(I) .EQ. 0.99999) T12(I) = 0
      IF(T13(I) .EQ. 0.99999) T13(I) = 0
      IF(T14(I) .EQ. 0.99999) T14(I) = 0
      IF(T15(I) .EQ. 0.99999) T15(I) = 0
      IF(T16(I) .EQ. 0.99999) T16(I) = 0
      IF(T17(I) .EQ. 0.99999) T17(I) = 0
      IF(T18(I) .EQ. 0.99999) T18(I) = 0
      IF(T21(I) .EQ. 0.99999) T21(I) = 0
      IF(T22(I) .EQ. 0.99999) T22(I) = 0
      IF(T23(I) .EQ. 0.99999) T23(I) = 0
      IF(T24(I) .EQ. 0.99999) T24(I) = 0
      IF(T25(I) .EQ. 0.99999) T25(I) = 0
      IF(T26(I) .EQ. 0.99999) T26(I) = 0
      IF(T27(I) .EQ. 0.99999) T27(I) = 0
      IF(T28(I) .EQ. 0.99999) T28(I) = 0
 10   CONTINUE
      TM(1) = TM(2) = TM(3) = 0
C
C  *****  CHANGE FROM DEGREES TO RADIANS *****
C
 20   DO 30 I=1,3
      WSTIR(I) = WST(I) * RPD
 30   ESTIR(I) = EST(I) * RPD
C
C  *****  SET UP SEAT INERTIA TENSOR  *****
C
      TINER(1,1) =  CMI(1)
      TINER(1,2) = -CPI(1)
      TINER(1,3) = -CPI(2)
      TINER(2,1) = -CPI(1)
      TINER(2,2) =  CMI(2)
      TINER(2,3) = -CPI(3)
      TINER(3,1) = -CPI(2)
      TINER(3,2) = -CPI(3)
      TINER(3,3) =  CMI(3)
C
C  CALCULATE THE DIRECTION COSINE MATRICES ......
C
      CALL DIRCOS (DES,ESTIR)
```

```
      CALL TRANS (DSE,DES,3,3)
C
C   *****  COMPUTE GRAVITY FORCES AND TORQUES *****
C
      DO 40 I=1,3
 40   FG(I) = CW * DES(I,3) * SSFLG
      CALL CRSPRD (TG,CCG,FG)
C
C   *****  SUM FORCES AND MOMENTS  *****
C
      DO 50 I=1,3
      F(I) = F11(I) + F12(I) + F13(I) + F14(I) + F15(I) + F16(I) +
     .       F17(I) + F18(I) + F21(I) + F22(I) + F23(I) + F24(I) +
     .       F25(I) + F26(I) + F27(I) + F28(I) + FG(I)
      T(I) = T11(I) + T12(I) + T13(I) + T14(I) + T15(I) + T16(I) +
     .       T17(I) + T18(I) + T21(I) + T22(I) + T23(I) + T24(I) +
     .       T25(I) + T26(I) + T27(I) + T28(I) + TG(I)
 50   CONTINUE
C
C  CALCULATE THE SEAT ALTITUDE ......
C
      ALT = -SRP(3)
C
C  CALCULATE THE DYNAMIC RESPONSE ......
C
      DR = SC * 86.977
C
C   ***************************************
C   *****  ANGULAR VELOCITY EQUATIONS  *****
C   ***************************************
C
C  CALCULATE TINER * WSTIR
C
      CALL MATMPY (TEMP1,TINER,WSTIR,3,3,1)
C
C  CALCULATE WSTIR X (TINER * WSTIR)
C
      CALL CRSPRD (TEMP2,WSTIR,TEMP1)
C
C  COMPUTE CCG X F...
C
      CALL CRSPRD (TEMP3,CCG,F)
C
C  SUM TERMS TO OBTAIN TOTAL TORQUE ......
C
      DO 60 I=1,3
 60   TEMP3(I) = T(I) - TEMP2(I) - TEMP3(I)
C
C  CALCULATE WDS ......
C
      CALL LUEQS (TINER,TEMP1,TEMP3,TEMP2,3,1,3,3,3,1.E-14,IERROR)
      IF(IERROR.NE.1) GO TO 80
      WRITE(6,70)
 70   FORMAT(* INERTIA MATRIX OF SEAT IS SINGULAR...RUN STOPPED*)
      STOP
 80   CONTINUE
C
      DO 90 I=1,3
```

```
      IF(IWS(I).NE.O) WDSIR(I) = TEMP1(I)
 90   WDS(I) = WDSIR(I) * DPR
C
C  ***********************************
C  *****  EULER ANGLE EQUATIONS  *****
C  ***********************************
C
      CALL EARATE (TEMP1,WSTIR,ESTIR)
      DO 100 I=1,3
 100  IF(IES(I).NE.0)EDS(I) = TEMP1(I) * DPR
C
C  ***************************************
C  *****  LINEAR VELOCITY EQUATIONS  *****
C  ***************************************
C
C  CALCULATE WDSTIR X CCG ......
C
      CALL CRSPRD (TEMP1,WDSIR,CCG)
C
C  CALCULATE WSTIR X CCG ......
C
      CALL CRSPRD (TEMP2,WSTIR,CCG)
C
C  CALCULATE WSTIR X (WSTIR X CCG) ......
C
      CALL CRSPRD (TEMP3,WSTIR,TEMP2)
C
C  CALCULATE WSTIR X UST ......
C
      CALL CRSPRD (TEMP2,WSTIR,UST)
C
C  CALCULATE F/M ......
C
      CMASS = CW/GRAV
      DO 120 I=1,3
 120  TEMP4(I) = F(I)/CMASS
C
C  SUM THE ACCELERATION COMPONENTS ......
C
      DO 130 I=1,3
 130  IF(IUS(I).NE.O) UDS(I) = TEMP4(I) - TEMP1(I) - TEMP2(I) - TEMP3(I)
C
C  =====  DETERMINE THE LOAD FACTORS  =====
C
      GX = (TEMP1(1) + TEMP3(1) - TEMP4(1))/GRAV
      GY = (TEMP1(2) + TEMP3(2) - TEMP4(2))/GRAV
      GZ = (TEMP1(3) + TEMP3(3) - TEMP4(3))/GRAV
C
C  ***************************************
C  *****  LINEAR POSITION EQUATIONS  *****
C  ***************************************
C
      CALL MATMPY (TEMP1,DSE,UST,3,3,1)
      DO 140 I=1,3
 140  IF(IXS(I).NE.O) XDS(I) = TEMP1(I)
C
C  ******************************************
C  *****  SPINAL COMPRESSION EQUATIONS  *****
```

```
C   ****************************************
C
C   SPINAL COMPRESSION VELOCITY EQUATION .....
C
      IF(ISCD.NE.0) SCDDOT = -23.6992 * SCD - 2798.41 * SC
     *                           + GRAV * GZ
C
C   SPINAL COMPRESSION EQUATION ......
C
      IF(ISC.NE.0) SCDOT = SCD
C
C   DURING TRIM, SUBTRACT TRIM VELOCITY FROM POSITION RATES
C
      IF (INST.NE.31) GO TO 160
      DO 150 I=1,3
 150  IF(IXS(I).NE.0) XDS(I)=XDS(I)-TM(I)
C
160   RETURN
      END
```

```
      SUBROUTINE SL (USL,USLD,IUSL,XSL,XSLD,IXSL,WSL,WSLD,IWSL,
     .                 ESL,ESLD,IESL,
     .                 UD,WD)
C
      DIMENSION USL(3),USLD(3),IUSL(3),XSL(3),XSLD(3),IXSL(3),
     .          WSL(3),WSLD(3),IWSL(3),ESL(3),ESLD(3),IESL(3),
     .          UD(3),WD(3)
C
      DIMENSION TEMP(3),WSLIR(3),ESLIR(3),DES(3,3),DSE(3,3)
C
      COMMON /CICCAL/ ICCAL
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
      DATA RPD,DPR / .01745329, 57.29578 /
C
C  DISIGNED BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C  EASIEST SIX DEGREE OF FREEDOM SLED MODEL
C
C  ==========  SLED OUTPUTS  ==========
C
C  LINEAR VELOCITIES - BODY AXIS
C
C     USL(3)  - X,Y,Z LINEAR VELOCITY VECTOR (FT/SEC)
C     USLD(3) - X,Y,Z LINEAR VELOCITY RATE VECTOR (FT/SEC/SEC)
C     IUSL(3) - INTEGRATION CONTROL
C
C  LINEAR POSITIONS - EARTH SYSTEM
C
C     XSL(3)  - X,Y,Z LINEAR POSITION VECTOR (FT)
C     XSLD(3) - X,Y,Z LINEAR POSITION RATE VECTOR (FT/SEC)
C     IXSL(3) - INTEGRATION CONTROL
C
C  ANGULAR VELOCITIES - BODY AXIS
C
C     WSL(3)  - X,Y,Z ANGULAR VELOCITY VECTOR - P,Q,R  (DEG/SEC)
C     WSLD(3) - X,Y,Z ANGULAR VELOCITY RATE VECTOR (DEG/SEC/SEC)
C     IWSL(3) - INTEGRATION CONTROL
C
C  EULER ANGLES  --  EARTH TO BODY  --  YAW,PITCH,ROLL
C
C     ESL(3)  - EARTH TO SLED EULER ANGLES (DEG)
C     ESLD(3) - EULER ANGLE RATES (DEG/SEC)
C     IESL(3) - INTEGRATION CONTROL
C
C  ==========  SLED INPUTS  ==========
C
C     UD(3)   - X,Y,Z SLED SYSTEM LINEAR VELOCITY RATE VECTOR (FT/SEC/SEC)
C     WD(3)   - X,Y,Z SLED SYSTEM ANGULAR VELOCITY RATE VECTOR (DEG/SEC/SEC)
C
C  ///////////////////////////////////////////////////////////////////////
C
C  ***************************
C  *****  INITIALIZATION  *****
C  ***************************
C
      IF(ICCAL.NE.1) GO TO 20
```

```
C
      DO 5 I=1,3
      IF(UD(I) .EQ. 0.99999) UD(I) = 0
 5    IF(WD(I) .EQ. 0.99999) WD(I) = 0
C
      IF(WSL(1)+WSL(2)+WSL(3).EQ.0) GO TO 20
C
      WRITE(6,10)
 10   FORMAT(/5X,*SLED ANGULAR VELOCITY IS NOT INITIALIZED AT ZERO *,
     .              * -----  RUN STOPPED  -----*//)
C
C ////////////////////////////////////////////////////////////////////////
C
C  CHANGE FROM DEGREES TO RADIANS ......
C
 20   DO 30 I=1,3
      WSLIR(I) = WSL(I) * RPD
 30   ESLIR(I) = ESL(I) * RPD
C
C  ******************************
C  *****  ANGULAR EQUATIONS  *****
C  ******************************
C
C  ANGULAR VELOCITY EQUATIONS ......
C
      DO 40 I=1,3
 40   IF(IWSL(I).NE.0) WSLD(I) = WD(I)
C
C  EULER ANGLE RATES ......
C
      CALL EARATE (TEMP,WSLIR,ESLIR)
      DO 50 I=1,3
  50   IF(IESL(I).NE.0) ESLD(I) = TEMP(I) * DPR
C
C  *****************************
C  *****  LINEAR EQUATIONS  *****
C  *****************************
C
C  LINEAR VELOCITY EQUATIONS ......
C
      CALL CRSPRD (TEMP,WSLIR,USL)
      DO 60 I=1,3
 60   IF(IUSL(I).NE.0) USLD(I) = UD(I) - TEMP(I)
C
C  LINEAR POSITION EQUATIONS ......
C
      CALL DIRCOS (DES,ESLIR)
      CALL TRANS (DSE,DES,3,3)
      CALL MATMPY (TEMP,DSE,USL,3,3,1)
      DO 70 I=1,3
 70   IF(IXSL(I).NE.0) XSLD(I) = TEMP(I)
C
      RETURN
      END
```

```
      SUBROUTINE SP (TRF,TMA,TST,
     .               WG,WGD,IWG,ESG,ESGD,IESG,ESR,ESRD,IESR,PHA,
     .               F,T,TIN,ECA,
     .               FL,YPR,AVW,WMI,SMI,RII,RIF,XR,UV,
     .               GSA,GSF,SPR,DPG,FMT,TMX,TNF,TOS,TSU,GMA,WST)
C
C   STANDARD COMPONENT SP CALCULATES FORCES AND TORQUES APPLIED
C   TO THE SEAT BY THE STAPAC STABILIZATION SYSTEM
C
C   ***************  SP TABLES  ***************
C
C      TRF  -  STAPAC ROCKET THRUST TABLE
C
C                THE INDEPENDENT VARIABLE IS TIME (SEC)
C                THE DEPENDENT VARIABLE IS ROCKET FORCE (LB)
C
C      TMA  -  MECHANICAL ADVANTAGE TABLE
C
C                THE INDEPENDENT VARIABLE IS THE GIMBAL ANGLE (DEG)
C                WITH RESPECT TO THE CAGED POSITION
C                THE DEPENDENT VARIABLE IS THE MECHANICAL ADVANTAGE
C
C      TST  -  SPRING TORQUE TABLE
C
C                THE INDEPENDENT VARIABLE IS THE GIMABAL ANGLE (DEG)
C                WITH RESPECT TO THE CAGED POSITION
C                THE DEPENDENT VARIABLE IS THE SPRING TORQUE (FT-LB)
C
C   ***************  SP OUTPUTS  ***************
C
C   ANGULAR VELOCITY  --  GIMBAL X-AXIS
C   (LESS THE SEAT ANGULAR VELOCITY PROJECTED UNTO THE GIMBAL X-AXIS)
C
C      WG  - ANGULAR VELOCITY (DEG/SEC)
C      WGD - ANGULAR ACCELERATION (DEG/SEC/SEC)
C      IWG - INTEGRATION CONTROL
C
C   EULER ANGLES  --  SEAT TO GIMBAL  --  YAW,PITCH,ROLL
C
C      ESG(3)  - SEAT TO GIMBAL EULER ANGLES (DEG)
C      ESGD(3) - EULER ANGLE RATES (DEG/SEC)
C      IESG(3) - INTEGRATION CONTROL
C
C   EULER ANGLES  --  SEAT TO ROCKET  --  YAW,PITCH,ROLL
C
C      ESR(3)  - ANGULAR POSITION (DEG)
C      ESRD(3) - ANGULAR VELOCITY (DEG/SEC)
C      IESR(3) - INTEGRATION CONTROL
C
C      PHA  - STAPAC OPERATIONAL PHASE
C                0 = BEFORE IGNITION
C                1 = STAPAC IGNITION
C                2 = STAPAC BURNOUT
C      F(3) - X,Y,Z SEAT BODY AXIS FORCE COMPONENTS (LB)
C      T(3) - X,Y,Z SEAT BODY AXIS TORQUE COMPONENTS (FT-LB)
C      TIN  - TIME AT STAPAC INITIATION (SEC)
C      ECA  - SEAT TO GIMBAL ROLL EULER ANGLE AT THE CAGED POSITION (DEG)
C
```

```
C  ***************  SP INPUTS  ***************
C
C     FL      - STAPAC IGNITION FLAG (1 = STAPAC ON)
C     YPR     - STAPAC APPLICATION FLAG
C                 1 = YAW STAPAC
C                 2 = PITCH STAPAC
C                 3 = ROLL STAPAC
C     AVW     - ANGULAR VELOCITY OF THE GYROSCOPE WHEEL (DEG/SEC)
C     WMI     - MOMENT OF INERTIA OF THE WHEEL ABOUT ITS
C               SPIN AXIS (SLUG-FT**2)
C     SMI     - MOMENT OF INERTIA OF THE SYSTEM LESS ROCKET ABOUT
C               THE GIMBAL AXIS (SLUG-FT**2)
C     RII     - MOMENT OF INERTIA OF THE ROCKET PRIOR TO
C               IGNITION (SLUG-FT**2)
C     RIF     - MOMENT OF INERTIA OF THE ROCKET AFTER
C               BURNOUT (SLUG-FT**2)
C     XR(3)   - X,Y,Z SEAT BODY AXIS POSITION VECTOR OF THE
C               ROCKET NOZZLE (FT)
C     UV(3)   - X,Y,Z ROCKET FORCE UNIT VECTOR IN THE ROCKET COORDINATE
C               SYSTEM
C     GSA     - GIMBAL MOTION STOP IN THE NEGATIVE ROLL DIRECTION WITH RESPECT
C               TO THE CAGED POSITION (DEG)
C     GSF     - GIMBAL MOTION STOP IN THE POSITIVE ROLL DIRECTION WITH RESPECT
C               TO THE CAGED POSITION (DEG)
C     SPR     - GIMBAL STOP ANGULAR RIGIDITY (FT-LB/DEG)
C     DPG     - GIMBAL STOP ANGULAR DAMPING (FT-LB/DEG/SEC)
C     FMT     - LOAD AT MAXIMUM FRICTION (LB)
C     TMX     - MAX FRICTION (FT-LB)
C     TNF     - FRICTION AT NO THRUST (FT-LB)
C     TOS     - THRUSTLINE OFFSET (LB)
C     TSU     - GYROSCOPE WHEEL SPINUP TIME (SEC)
C     GMA     - GIMBAL ANGULAR VELOCITY AT MAXIMUM FRICTION (DEG/SEC)
C     WST(3)  - X,Y,Z SEAT BODY AXIS ANGULAR VELOCITY VECTOR OF
C               THE SEAT (DEG/SEC)
C
C  ****************************************************
C
C  ----  DIMENSIONS OF CALLING ARGUMENTS  ----
C
      DIMENSION TRF(5),TMA(5),TST(5),ESG(3),ESGD(3),IESG(3),
     *          ESR(3),ESRD(3),IESR(3),F(3),T(3),XR(3),UV(3),
     *          WST(3)
C
C  ----  INTERNAL DIMENSIONS  ----
C
      DIMENSION WGB(3),WRB(3),FRKT(3),WSTG(3),TEMP(3),DSR(3,3),DRS(3,3),
     *          DSG(3,3),ESRIR(3),ESGIR(3)
C
      COMMON /CICCAL/ ICCAL
      COMMON /CTIME/ TIME
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
      DATA RPD,DPR / .01745329, 57.29578 /
      DATA PI / 3.14159 /
      DATA WGB(2),WGB(3) / 0 , 0 /, WRB(1),WRB(3) / 0 , 0 /
C
C  ***************************
C  *****  INITIALIZATION  *****
```

```
C   ****************************
C
      IF(ICCAL.NE.1) GO TO 20
C
      PHA = TIN = 0
      ECA = ESG(3)
      DO 10 I=1,3
      F(I) = 0.
 10   T(I) = 0.
      IF(TSU.EQ.0.99999) TSU = 0.005
      IF(GMA.EQ.0.99999) GMA = 10.
      IF(UV(1).EQ.0.99999) UV(1) = 0
      IF(UV(2).EQ.0.99999) UV(2) = 0
      IF(UV(3).EQ.0.99999) UV(3) = -1.
C
C   ////////////////////////////////////////////////////////////
C
C   BYPASS COMPONENT IF STAPAC IS OFF ......
C
 20    IF(FL.NE.1. .OR. PHA.EQ.2.) GO TO 260
C
C   ----  WRITE IGNITION MESSAGE AND INITIALIZE START TIME  ----
C
      IF(PHA.EQ.1.) GO TO 90
C
      IF(YPR.EQ.2.) GO TO 40
      IF(YPR.EQ.3.) GO TO 60
      WRITE(6,30)  TIME
 30   FORMAT(/5X,*YAW STAPAC IGNITION AT TIME=*,F10.4,2X,*SEC*/)
      GO TO 80
 40   WRITE(6,50)  TIME
 50   FORMAT(/5X,*PITCH STAPAC IGNITION AT TIME=*,F10.4,2X,*SEC*/)
      GO TO 80
 60   WRITE(6,70)  TIME
 70   FORMAT(/5X,*ROLL STAPAC IGNITION AT TIME=*,F10.4,2X,*SEC*/)
 80   TIN = TIME
      PHA = 1.
 90   CONTINUE
C
C   ----  CHANGE FROM DEGREES TO RADIANS  ----
C
      DO 100 I=1,3
      ESGIR(I) = ESG(I) * RPD
 100  ESRIR(I) = ESR(I) * RPD
C
C   ----  COMPUTE THE SEAT TO GIMBAL DIRECTION COSINE MATRIX  ----
C
      CALL DIRCOS (DSG,ESGIR)
C
C   ----  CALCULATE THE SEAT ANGULAR VELOCITY IN THE GIMBAL SYSTEM  ----
C
      CALL MATMPY (WSTG,DSG,WST,3,3,1)
C
C   ----  DETERMINE THE TIME INTO STAPAC  ----
C
      TIS = TIME - TIN
C
C   ----  DETERMINE THE ROCKET THRUST  ----
```

```
C
      NRT = TRF(2)
      IF(TIS.GT.TRF(NRT+3)) GO TO 190
      FR = TBLU1 (TIS,TRF(4),TRF(NRT+4),1,-NRT)
C
C  ----  DETERMINE THE MECHANICAL ADVANTAGE  ----
C
      DELTA = ESG(3) - ECA
C
      NMA = TMA(2)
      SMA = TBLU1 (DELTA,TMA(4),TMA(NMA+4),1,-NMA)
C
C  ----  CALCULATE THE SYSTEM INERTIA  ----
C
      SYSMI = SMI + SMA**2*(RII-(TIS-TRF(4))/(TRF(NRT+3)-TRF(4))
     .                       *(RII-RIF))
C
C  *********************************************
C  *****  DETERMINE THE GIMBAL X-AXIS TORQUE *****
C  *********************************************
C
C  ----  CALCULATE THE THRUSTLINE OFFSET TORQUE  ----
C
      TOFF = FR * TOS * SMA
C
C  ----  CALCULATE THE FRICTIONAL TORQUE  ----
C
      ANGV = (WG - WSTG(1))/GMA
      TFRICT = -SIGN(AMIN1(1.,ABS(ANGV)),ANGV) * ABS(SMA) *
     .          (TNF+FR/FMT*(TMX-TNF))
C
C  ----  CALCULATE THE PRECESSIONAL TORQUE  ----
C
      AVWIR = AVW * RPD
      IF(TIS.LE.TSU) AVWIR = (1.+SIN(3.*PI/2.+TIS/TSU*PI))/2.*AVW*RPD
      IF(TIS.LE.0) AVWIR = 0
      TPREC = -WMI * AVWIR * WSTG(2) * RPD
C
C  ----  DETERMINE THE SPRING TORQUE  ----
C
      NST = TST(2)
      TSPR = TBLU1 (DELTA,TST(4),TST(NST+4),1,-NST)
C
C  ----  CALCULATE THE GYMBAL STOP TORQUE  ----
C
      IF(DELTA.LT.GSA) GO TO 110
      IF(DELTA.GT.GSF) GO TO 120
      GO TO 140
C
C  CALCULATE SPRING TORQUE ......
C
 110  TSTOP = SPR * (GSA - DELTA)
      GO TO 130
 120  TSTOP = SPR * (GSF - DELTA)
C
C  CALCULATE DAMPING TORQUE ......
C
 130  TSTOP = TSTOP - ANGV * DPG
```

```
      GO TO 150
C
C  SET SPRING AND DAMPING TORQUES EQUAL TO ZERO ....
C
 140  TSTOP = 0.
C
C  ----  SUM THE TORQUES  ----
C
 150  TSUM = TOFF + TFRICT + TPREC + TSPR + TSTOP
C
C  ********************************
C  *****  CALCULATE THE RATES  *****
C  ********************************
C
C  ----  CALCULATE THE GIMBAL X-AXIS ANGULAR VELOCITY RATE  ----
C
      IF(IWG.NE.0) WGD = (TSUM/SYSMI) * DPR
C
C  ----  DETERMINE THE GIMBAL EULER ANGLE RATES  ----
C
      WGB(1) = WG - WSTG(1)
      CALL EARATE (TEMP,WGB,ESGIR)
      DO 160 I=1,3
 160  IF(IESG(I).NE.0) ESGD(I) = TEMP(I)
C
C  ----  COMPUTE THE ROCKET EULER ANGULAR RATES ----
C
      WRB(2) = WGB(1) * SMA
      CALL EARATE (TEMP,WRB,ESRIR)
      DO 170 I=1,3
 170  IF(IESR(I).NE.0) ESRD(I) = TEMP(I)
C
C  ************************************************
C  *****  CALCULATE THE ROCKET FORCES AND TORQUES  *****
C  ************************************************
C
C  ----  TRANSFORM THE ROCKET THRUST TO THE SEAT  ----
C
      CALL DIRCOS (DSR,ESRIR)
      CALL TRANS (DRS,DSR,3,3)
      DO 180 I=1,3
 180  FRKT(I) = UV(I) * FR
      CALL MATMPY (F,DRS,FRKT,3,3,1)
C
C  ----  COMPUTE THE SEAT BODY AXIS TORQUE COMPONENTS  ----
C
      CALL CRSPRD (T,XR,F)
      GO TO 260
C
C  ****************************************
C  *****  WHEN THE ROCKET SHUTS DOWN ....  *****
C  ****************************************
C
C  ----  ZERO OUT RATES, FORCES, AND TORQUES  ----
C
 190  DO 200 I=1,3
      ESGD(I) = 0.
      ESRD(I) = 0.
```

```
      F(I) = 0.
  200 T(I) = 0.
      WGD = 0.
      PHA = 2.
C
C  ----  WRITE BURNOUT MESSAGE  ----
C
      IF(YPR.EQ.2.) GO TO 220
      IF(YPR.EQ.3.) GO TO 240
      WRITE (6,210) TIME
  210 FORMAT(/5X,*YAW STAPAC BURNOUT AT TIME=*,F10.4,2X,*SEC*/)
      GO TO 260
  220 WRITE(6,230) TIME
  230 FORMAT(/5X,*PITCH STAPAC BURNOUT AT TIME=*,F10.4,2X,*SEC*/)
      GO TO 260
  240 WRITE(6,250) TIME
  250 FORMAT(/5X,*ROLL STATPAC BURNOUT AT TIME=*,F10.4,2X,*SEC*/)
C
  260 RETURN
      END
```

```
      SUBROUTINE SR (TRF,
     .               PW,PWD,IPW,
     .               PHA,RON,FST,TST,XCG,PMI,PPI,FR,PWI,SPI,RHO,
     .               VWI,TMI,TIG,
     .               FON,PCG,EA,XRN,YAW,PIT,PL,POD,PID)
C
C  FORCES AND MOMENTS ACTING ON THE SEAT FROM THE SUSTAINER ROCKET
C
C  DESIGNED BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C  ***************  ROCKET TABLES  ***************
C
C      TRF - ROCKET THRUST TABLE
C
C              THE INDEPENDENT VARIABLE IS TIME (SEC)
C              THE DEPENDENT VARIABLE IS THE ROCKET FORCE (LB)
C
C  ***************  ROCKET OUTPUTS  ***************
C
C      PW  - WEIGHT OF UNBURNED PROPELLANT (LB)
C      PWD - PROPELLANT BURN RATE (LB/SEC)
C      IPW - INTEGRATION CONTROL
C
C      PHA    - ROCKET PHASE
C                 0 = BEFORE IGNITION
C                 1 = ROCKET BURN
C                 2 = ROCKET OFF
C      RON    - ROCKET ON FLAG (1=ON  0=OFF)
C      FST(3) - X,Y,Z SEAT SYSTEM ROCKET FORCE COMPONENTS (LB)
C      TST(3) - X,Y,Z SEAT SYSTEM ROCKET TORQUE COMPONENTS (FT-LB)
C      XCG(3) - X,Y,Z SEAT SYSTEM POSITION VECTOR OF THE
C               PROPELLANT CENTER OF GRAVITY (FT)
C      PMI(3) - PROPELLANT MOMENTS OF INERTIA - IXX,IYY,IZZ (SLUG-FT**2)
C      PPI(3) - PROPELLANT PRODUCTS OF INERTIA - IXY,IXZ,IYZ (SLUG-FT**2)
C      FR     - SUSTAINER ROCKET FORCE MAGNITUDE (LB)
C      PWI    - INITIAL WEIGHT OF THE PROPELLANT (LB)
C      SPI    - ROCKET PROPELLANT SPECIFIC IMPULSE (LB-SEC/LB)
C      RHO    - ROCKET PROPELLANT DENSITY (LB/FT**3)
C      VWI    - INITIAL VIRTUAL WEIGHT (LB)
C      TMI(3) - SOLID GRAIN MOMENTS OF INERTIA - IXX,IYY,IZZ (SLUG-FT**2)
C      TIG    - ROCKET IGNITION TIME (SEC)
C
C  ***************  ROCKET INPUTS  ***************
C
C      FON    - ROCKET ON FLAG (1=ON)
C      PCG(3) - INITIAL X,Y,Z SEAT SYSTEM POSTION VECTOR OF THE
C               PROPELLANT CENTER OF GRAVITY (FT)
C      EA(3)  - SEAT TO ROCKET PROPELLANT EULER ANGLES (DEG)
C      XRN(3) - X,Y,Z PROPELLANT SYSTEM POSITION VECTOR OF THE ROCKET
C               NOZZLE (FT)
C      YAW    - YAW EULER ANGLE OF THE THRUST VECTOR IN THE PROPELLANT
C               COORDINATE SYSTEM (DEG)
C      PIT    - PITCH EULER ANGLE OF THE THRUST VECTOR IN THE PROPELLANT
C               COORDINATE SYSTEM (DEG)
C      PL     - PROPELLANT GRAIN LENGTH (FT)
C      POD    - PROPELLANT GRAIN OUTSIDE DIAMETER (FT)
C      PID    - PROPELLANT GRAIN INSIDE DIAMETER (FT)
```

```
C
C  DIMENSIONS OF CALLING ARGUMENTS ......
C
      DIMENSION TRF(5),FST(3),TST(3),XCG(3),PMI(3),PPI(3),
     .          TMI(3),PCG(3),EA(3),XRN(3)
C
C  INTERNAL DIMENSIONS ......
C
      DIMENSION VMIN(3),DSP(3,3),DPS(3,3),DC(3),EAIR(3),TEMP(3),
     .          XRNST(3)
C
      COMMON /CTIME/ TIME
      COMMON /CICCAL/ ICCAL
      COMMON /COVRLY/ INST
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
      DATA RPD /.01745329/
      DATA GRAV /32.174/
C
C  ***************************
C  *****  INITIALIZATION  *****
C  ***************************
C
      IF(ICCAL.NE.1) GO TO 80
C
C  DEFINE THE PROPELLANT CENTER OF GRAVITY IN THE SEAT SYSTEM
C  FOR OUTPUT ......
C
      DO 10 I=1,3
 10   XCG(I) = PCG(I)
C
C  MISC INITIALIZATION ......
C
      IF(PW.NE.0) GO TO 30
      WRITE(6,20)
 20   FORMAT(/5X,* ====  PROPELLANT WEIGHT NOT INITIALIZED - RUN*,
     .          * STOPPED  ==== */)
      STOP
C
 30   PWI = PW
      PHA = RON = FR = TIG = 0
C
      DO 40 I=1,3
 40   PPI(I) = FST(I) = TST(I) = 0
C
C  CALCULATE THE SUSTAINER ROCKET'S TOTAL IMPULSE ......
C
      TOTIMP = 0
      NA = TRF(2)
      DO 50 I=2,NA
      DELIMP = (TRF(I+NA+3)-TRF(I+NA+2))/(TRF(I+3)-TRF(I+2))*0.5
 50   TOTIMP = TOTIMP + DELIMP
C
C  CALCULATE THE SPECIFIC IMPULSE ......
C
      SPI = TOTIMP/PW
C
C  CALCULATE THE INITIAL GRAIN VOLUME ......
```

```
C
      PV = 0.7854 * PL * (POD**2 - PID**2)
C
C  CALCULATE THE DENSITY ......
C
      RHO = PW/PV
C
C  INITIAL VIRTUAL WEIGHT (THE EMPTY PORTION OF THE GRAIN) ......
C
      VWI = 0.7854 * PL * PID**2 * RHO
C
C  VIRTUAL MASS MOMENTS OF INERTIA ......
C
      VMIMASS = VWI/GRAV
      VMIN(1) = (VMIMASS/12.) * (3.*(PID/2.)**2 + PL**2)
      VMIN(2) = VMIN(1)
      VMIN(3) = (VMIMASS/2.) * (PID/2.)**2
C
C  TOTAL MASS AS IF IT WERE A COMPLETELY SOLID GRAIN ......
C
      TMASS = 0.7854 * POD**2 * PL * RHO/GRAV
C
C  TOTAL MOMENT OF INERTIAS ......
C
      TMI(1) = (TMASS/12.) * (3.*(POD/2.)**2 + PL**2)
      TMI(2) = TMI(1)
      TMI(3) = (TMASS/2.) * (POD/2.)**2
C
C  INITIAL PROPELLANT MOMENT OF INERTIAS ......
C
      DO 60 I=1,3
 60   PMI(I) = TMI(I) - VMIN(I)
C
C  ROTATE THE PROPELLENT INERTIAS INTO THE SEAT SYSTEM ......
C
      DO 70 I=1,3
 70   EAIR(I) = EA(I) * RPD
      CALL DIRCOS (DSP,EAIR)
      CALL TRANS (DPS,DSP,3,3)
      CALL ROTATEI (PMI,PPI,DPS)
C
C  ////////////////////////////////////////////////////////////
C
C  RETURN IF SUSTAINER ROCKET IS OFF ......
C
 80   IF(FON.EQ.0 .OR. PHA.EQ.2.) GO TO 160
C
C  ======  ROCKET ON  ======
C
      IF(PHA.EQ.1.) GO TO 100
      PHA = RON = 1.
      TIG = TIME
C
      IF(INST.EQ.26) WRITE(6,90) TIME
 90   FORMAT(/5X,*SUSTAINER ROCKET ON AT TIME = *,F10.4,* SEC*/)
C
C  COMPUTE THE DIRECTION COSINE MATRICES ......
C
```

```
 100  DO 110 I=1,3
 110  EAIR(I) = EA(I) * RPD
      CALL DIRCOS (DSP,EAIR)
      CALL TRANS (DPS,DSP,3,3)
C
C  COMPUTE THRUST VECTOR DIRECTION COSINES ......
C
      THE = PIT * RPD
      PSI = YAW * RPD
      DC(1) = COS(PSI) * SIN(THE)
      DC(2) = SIN(PSI) * SIN(THE)
      DC(3) = COS(THE)
C
C  CALCULATE THE BODY AXIS FORCE AND TORQUE COMPONENTS ......
C
      NA = TRF(2)
      TINRKT = TIME - TIG
      IF(TINRKT.GE.TRF(NA+3)) GO TO 130
      FR = -TBLU1(TINRKT,TRF(4),TRF(NA+4),1,-NA)
      DO 120 I=1,3
 120  TEMP(I) = DC(I) * FR
      CALL MATMPY (FST,DPS,TEMP,3,3,1)
      CALL VECXYZ (XRNST,XRN,PCG,DPS,2)
      CALL CRSPRD (TST,XRNST,FST)
C
C  PROPELLANT CONSUMPTION RATE (LB/SEC).....
C
      IF(IPW.NE.0) PWD = -FR/SPI
C
C  PROPELLANT MASS BURNED (SLUGS).....
C
      BM = (PWI - PW)/GRAV
C
C  BURNED VOLUME (FT**3).....
C
      BVOL = BM/RHO/GRAV
C
C  BURNED RADIUS OF GRAIN (FT).....
C
      BR = SQRT((BVOL/PL/0.7854) + PID**2)/2.0
C
C  NEW VIRTUAL MASS (SLUGS).....
C
      VMP = VWI/GRAV + BM
C
C  NEW VIRTUAL INERTIAS.....
C
      VMIN(1) = (VMP/12.) * (3.*BR**2 + PL**2)
      VMIN(2) = VMIN(1)
      VMIN(3) = (VMP/2.) * BR**2
C
C  INERTIAS OF REMAINING PROPELLANT.....
C
      PMI(1) = TMI(1) - VMIN(1)
      PMI(2) = PMI(1)
      PMI(3) = TMI(3) - VMIN(3)
C
C  ROTATE THE ROCKET PROPELLANT INERTIA PROPERTIES INTO THE
```

```
C  SEAT AXIS SYSTEM ......
C
      CALL ROTATE1 (PMI,PPI,DPS)
      GO TO 160
C
C  ======  AT ROCKET BURNOUT ......  ======
C
 130  IF(INST.EQ.26) WRITE(6,140) TIME
 140  FORMAT(/5X,*SUSTAINER ROCKET OFF AT TIME=*,F10.4,2X,*SEC*/)
      PWD = RON = 0
      PHA = 2.
      DO 150 I=1,3
 150  FST(I) = TST(I) = PMI(I) = 0
C
 160  RETURN
      END
```

```
      SUBROUTINE WB (CW,CCG,CMI,CPI,
     .               AB,WS,XS,SMI,SPI,W1,X1,BM1,BP1,
     .               W2,X2,BM2,BP2,W3,X3,BM3,BP3)
C
      DIMENSION CCG(3),CMI(3),CPI(3),
     .          XS(3),SMI(3),SPI(3),X1(3),BM1(3),BP1(3),
     .          X2(3),BM2(3),BP2(3),X3(3),BM3(3),BP3(3)
C
      DIMENSION TSMI(3),T1MI(3),T2MI(3),T3MI(3),
     .          TSPI(3),T1PI(3),T2PI(3),T3PI(3),DIFF(3)
C
      COMMON /CICCAL/ ICCAL
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
      DATA GRAV /32.174/
C
C  DESIGNED BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C  NOTE -  ALL MOMENT AND PRODUCT OF INERTIA VECTORS INPUT
C          INTO THIS ROUTINE HAVE BEEN ROTATED INTO THE
C          SEAT COORDINATE SYSTEM.
C
C  ***************  WB OUTPUTS  ***************
C
C     CW     - COMPOSITE SEAT WEIGHT (LB)
C     CCG(3) - X,Y,Z SEAT AXIS SYSTEM COMPOSITE CENTER OF GRAVITY (FT)
C     CMI(3) - COMPOSITE SEAT MOMENT OF INERTIA VECTOR ABOUT ITS
C              CENTER OF GRAVITY - IXX,IYY,IZZ (SLUG-FT**2)
C     CPI(3) - COMPOSITE SEAT PRODUCT OF INERTIA VECTOR ABOUT ITS
C              CENTER OF GRAVITY - IXX,IYY,IZZ (SLUG-FT**2)
C
C  ***************  WB INPUTS  ***************
C
C     AB     - NUMBER OF BODIES ATTACHED TO THE BASIC SEAT
C     WS     - BASIC SEAT WEIGHT (LB)
C     XS(3)  - X,Y,Z SEAT AXIS SYSTEM POSTION VECTOR OF THE
C              BASIC SEAT CENTER OF GRAVITY (FT)
C     SMI(3) - MOMENT OF INERTIA VECTOR FOR THE BASIC SEAT ABOUT
C              ITS CENTER OF GRAVITY - IXX,IYY,IZZ (SLUG-FT**2)
C     SPI(3) - PRODUCT OF INERTIA VECTOR FOR THE BASIC SEAT ABOUT
C              ITS CENTER OF GRAVITY - IXY,IXZ,IYZ (SLUG-FT**2)
C     W1     - WEIGHT OF BODY ONE (LB)
C     X1(3)  - X,Y,Z SEAT AXIS SYSTEM POSITION VECTOR OF THE
C              CENTER OF GRAVITY FOR BODY ONE (FT)
C     BM1(3) - MOMENT OF INERTIA VECTOR FOR BODY ONE ABOUT ITS
C              CENTER OF GRAVITY - IXX,IYY,IZZ (SLUG-FT**2)
C     BP1(3) - PRODUCT OF INERTIA VECTOR FOR BODY ONE ABOUT ITS
C              CENTER OF GRAVITY - IXY,IXZ,IYZ (SLUG-FT**2)
C     W2     - WEIGHT OF BODY TWO (LB)
C     X2(3)  - X,Y,Z SEAT AXIS SYSTEM POSITION VECTOR OF THE
C              CENTER OF GRAVITY FOR BODY TWO (FT)
C     BM2(3) - MOMENT OF INERTIA VECTOR FOR BODY TWO ABOUT ITS
C              CENTER OF GRAVITY - IXX,IYY,IZZ (SLUG-FT**2)
C     BP2(3) - PRODUCT OF INERTIA VECTOR FOR BODY TWO ABOUT ITS
C              CENTER OF GRAVITY - IXY,IXZ,IYZ (SLUG-FT**2)
C     W3     - WEIGHT OF BODY THREE (LB)
C     X3(3)  - X,Y,Z SEAT AXIS SYSTEM POSITON VECTOR OF THE
```

```
C              CENTER OF GRAVITY FOR BODY THREE (FT)
C     BM3(3) - MOMENT OF INERTIA VECTOR FOR BODY THREE ABOUT ITS
C              CENTER OF GRAVITY - IXX,IYY,IZZ (SLUG-FT**2)
C     BPI(3) - PRODUCT OF INERTIA VECTOR FOR BODY THREE ABOUT ITS
C              CENTER OF GRAVITY - IXY,IXZ,IYZ (SLUG-FT**2)
C
C  ****************************
C  *****  INITIALIZATION  *****
C  ****************************
C
      IF(ICCAL.NE.1) GO TO 80
C
      IF(AB.EQ.0.99999) AB = 0.
C
C  ZERO WEIGHTS AND INERTIAS OF NON-EXISTANT BODIES
C
      IF(W1 .EQ. 0.99999) W1 = 0.
      IF(W2. EQ. 0.99999) W2 = 0.
      IF(W3 .EQ. 0.99999) W3 = 0.
      IF(AB.GE.1.) GO TO 20
      DO 10 I=1,3
      IF(X1(I).EQ..99999)X1(I)=0
      IF(BM1(I).EQ..99999)BM1(I)=0
 10   IF(BP1(I).EQ..99999)BP1(I)=0
 20   IF(AB.GE.2.) GO TO 40
      DO 30 I=1,3
      IF(X2(I).EQ..99999)X2(I)=0
      IF(BM2(I).EQ..99999)BM2(I)=0
 30   IF(BP2(I).EQ..99999)BP2(I)=0
 40   IF(AB.GE.3.) GO TO 60
      DO 50 I=1,3
      IF(X3(I).EQ..99999)X3(I)=0
      IF(BM3(I).EQ..99999)BM3(I)=0
 50   IF(BP3(I).EQ..99999)BP3(I)=0
C
C  ZERO OUT THE MOMENT AND PRODUCT VECTORS ......
C
 60   DO 70 I=1,3
      TSMI(I) = T1MI(I) = T2MI(I) = T3MI(I) = 0
 70   TSPI(I) = T1PI(I) = T2PI(I) = T3PI(I) = 0
C
C  ///////////////////////////////////////////////////////////////
C
C  COMPUTE THE LOCATION OF THE COMPOSITE C.G. FROM THE SRP ......
C
 80   CW = WS + W1 + W2 + W3
      DO 90 I=1,3
 90   CCG(I) = (WS*XS(I) + W1*X1(I) + W2*X2(I) + W3*X3(I))/CW
C
C  **************************************
C  *****  SEAT INERTIA PROPERTIES  *****
C  **************************************
C
C  CALCULATE THE SEAT MASS ......
C
      BMASS = WS/GRAV
C
C  COMPUTE THE INERTIA PROPERTIES ......
```

```
C
      DO 100 I=1,3
  100 DIFF(I) = CCG(I) - XS(I)
      CALL PAXIS (TSMI,TSPI,SMI,SPI,BMASS,DIFF)
C
C   ***************************************
C   *****   BODY 1 INERTIA PROPERTIES   *****
C   ***************************************
C
      IF (AB.LT.1.0) GO TO 140
C
C   CALCULATE THE MASS OF BODY 1 ......
C
      BMASS = W1/GRAV
C
C   COMPUTE THE INERTIA PROPERTIES ......
C
      DO 110 I=1,3
  110 DIFF(I) = CCG(I) - X1(I)
      CALL PAXIS (T1MI,T1PI,BM1,BP1,BMASS,DIFF)
C
C   ***************************************
C   *****   BODY 2 INERTIA PROPERTIES   *****
C   ***************************************
C
      IF (AB.LT.2.0) GO TO 140
C
C   CALCULATE THE MASS OF BODY 2 ......
C
      BMASS = W2/GRAV
C
C   COMPUTE THE INERTIA PROPERTIES ......
C
      DO 120 I=1,3
  120 DIFF(I) = CCG(I) -X2(I)
      CALL PAXIS (T2MI,T2PI,BM2,BP2,BMASS,DIFF)
C
C   ***************************************
C   *****   BODY 3 INERTIA PROPERTIES   *****
C   ***************************************
C
      IF (AB.LT.3.0) GO TO 140
C
C   CALCULATE THE MASS OF BODY 3 ......
C
      BMASS = W3/GRAV
C
C   COMPUTE THE INERTIA PROPERTIES ......
C
      DO 130 I=1,3
  130 DIFF(I) = CCG(I) - X3(I)
      CALL PAXIS (T3MI,T3PI,BM3,BP3,BMASS,DIFF)
C
C   *******************************************************
C   *****   COMPUTE THE COMPOSITE BODY INERTIA PROPERTIES   *****
C   *******************************************************
C
  140 DO 150 I=1,3
```

```
      CMI(I) = TSMI(I) + T1MI(I) + T2MI(I) + T3MI(I)
  150 CPI(I) = TSPI(I) + T1PI(I) + T2PI(I) + T3PI(I)
C
      RETURN
      END
```

APPENDIX H

EASIEST SUBROUTINES AND FUNCTIONS

This appendix contains listings of the EASIEST subroutines and functions, which include the following:

ARTAN2	DISECT	PAXIS
BLOCK	EARATE	PCAERO
BRIDL2	FSW	RATIO
BRIDL3	LAG	RLIM
BRIDL4	LIBRIDL	ROTATEI
CAD	LILINE	TBLU3
CEAERO	LILOAD	TLU
COSDIR	LINDST	UNPACK
DET3	LINEPL	VECXYZ
DIRCOS	LOOK	VELXYZ

```
      FUNCTION ARTAN2(AI,AR)
C
C  FOUR QUADRANT ARCTANGENT FUNCTION, AI BEING THE NUMERATOR AND AR
C  BEING THE DENOMINATOR.
C
      IF(ABS(AR) - .000001 * ABS(AI)) 10,10,30
 10   IF(AI) 20,50,20
 20   ARTAN2 = 1.55079 * SIGN(1.,AI)
      GO TO 60
C
 30   ARTAN2 = ATAN(AI/AR)
      IF(AR) 40,20,60
 40   ARTAN2 = 3.14159 + ARTAN2
      N = ARTAN2/3.14159
      EN = N
      ARTAN2 = ARTAN2 - 6.28318*EN
      GO TO 60
C
 50   ARTAN2 = 0.
 60   RETURN
      END
```

```
      SUBROUTINE BLOCK (FSEAT,FAIRP,TSEAT,TAIRP,ZBL,
     .                  SRPR,SRPA,RAILL,XRAIL,XBS,SPR,DPG,FRICT,
     .                  UST,WST,UAP,WAP,DAE,DER,DRS,DRA,DSA,DSE,
     .                  DSR,UP,FLAG)
C
C  DESIGNED BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C  CALCULATES THE FORCES AND MOMENTS ON THE SEAT AND AIRPLANE FROM
C  THE BLOCKS
C
C  ***************  BLOCK OUTPUTS  ***************
C
C     FSEAT(3)   - X,Y,Z SEAT BODY AXIS FORCE COMPONENTS (LB)
C     FAIRP(3)   - X,Y,Z AIRPLANE BODY AXIS FORCE COMPONENTS (LB)
C     TSEAT(3)   - X,Y,Z SEAT BODY AXIS MOMENT COMPONENTS (FT-LB)
C     TAIRP(3)   - X,Y,Z AIRPLANE BODY AXIS MOMENT COMPONENTS (FT-LB)
C     ZBL        - RAIL AXIS Z COORDINATE OF BLOCK
C
C  ***************  BLOCK INPUTS  ***************
C
C     SRPR(3)    - X,Y,Z RAIL POSITION VECTOR OF THE SRP (FT)
C     SRPA(3)    - X,Y,Z AIRPLANE POSITION VECTOR OF THE SRP (FT)
C     RAILL      - RAIL LENGTH (FT)
C     XRAIL      - X,Y,Z AIRPLANE POSITION VECTOR OF THE ORIGIN
C                  OF THE RAIL COORDINATE SYSTEM (FT)
C                  POINT ON THE AIRPLANE (FT)
C     XBS(3)     - X,Y,Z SEAT POSTION VECTOR OF THE BLOCK (FT)
C     SPR        - RAIL SPRING CONSTANT (LB/FT)
C     DPG        - RAIL DAMPING CONSTANT (LB/FT/SEC)
C     FRICT      - SLIDER BLOCK FRICTION COEFFICEINT
C     UST(3)     - SEAT AXIS VELOCITY OF SEAT REFERENCE POINT (FT/SEC)
C     WST(3)     - SEAT AXIS ANGULAR RATES OF SEAT (RAD/SEC)
C     UAP(3)     - AIRPLANE AXIS VELOCITY OF AIRPLANE (FT/SEC)
C     WAP(3)     - AIRPLANE AXIS ANGULAR RATES OF AIRPLANE (RAD/SEC)
C     DAE(3,3)   - AIRPLANE TO EARTH DIRECTION COSINE MATRIX
C     DER(3,3)   - EARTH TO RAILS DIRECTION COSINE MATRIX
C     DRS(3,3)   - RAILS TO SEAT DIRECTION COSINE MATRIX
C     DRA(3,3)   - RAILS TO AIRPLANE DIRECTION COSINE MATRIX
C     DSA(3,3)   - SEAT TO AIRPLANE DIRECTION COSINE MATRIX
C     DSE(3,3)   - SEAT TO EARTH DIRECTION COSINE MATRIX
C     DSR(3,3)   - SEAT TO RAILS DIRECTION COSINE MATRIX
C     UP         - EJECTION DIRECTION FLAG
C                        +1 = UPWARD WRT THE AIRPLANE
C                        -1 = DOWNWARD WRT THE AIRPLANE
C     FLAG       - BLOCK POSITION SWITCH ( 0 = ON RAILS   1 = OFF RAILS )
C
C  DIMENSIONS OF CALLING ARGUMENTS ......
C
      DIMENSION FSEAT(3),FAIRP(3),TSEAT(3),TAIRP(3),SRPR(3),SRPA(3),
     .          XRAIL(3),XBS(3),SPR(2),DPG(2),UST(3),WST(3),UAP(3),
     .          WAP(3),DAE(3,3),DER(3,3),DRS(3,3),DRA(3,3),DSA(3,3),
     .          DSE(3,3),DSR(3,3)
C
C  INTERNAL DIMENSIONS ......
C
      DIMENSION XBR(3),USBE(3),FDEFL(3),ARM(3),
     .          XBA(3),UABE(3),RVBE(3),RVBR(3),TEMP(3)
```

```
C
      COMMON/COVRLY/INST
      COMMON/CIO/IREAD,IWRITE,IDIAG
      DATA TEMP /0,0,0/
C
C  CALCULATION OF SLIDER BLOCK LOCATION IN THE RAIL AXIS SYSTEM ......
C
      CALL VECXYZ (XBR,XBS,SRPR,DSR,2)
      ZBL = XBR(3)
      TEMP(3) = XBR(3)
C
C  SET FORCES = 0 IF BLOCK OFF RAILS (EXCEPT DURING INITIALIZATION) ......
C
      FLAG = 0
      IF(INST.EQ.31.OR.INST.EQ.61) GO TO 20
      IF(XBR(3)*UP.GT.RAILL*UP) GO TO 20
      DO 10 I=1,3
      FSEAT(I)=0.
      FAIRP(I)=0.
      TSEAT(I)=0.
      TAIRP(I)=0.
 10   CONTINUE
      FLAG = 1.
      GO TO 50
C
C  COMPUTE VELOCITY OF BLOCK IN EARTH AXES SYSTEM ......
C
 20   CALL VELXYZ (USBE,UST,XBS,WST,DSE)
C
C  COORDINATES OF BLOCK IN AIRPLANE AXES SYSTEM ......
C
      CALL VECXYZ (XBA,XBS,SRPA,DSA,2)
C
C  VELOCITY OF BLOCK POSITION WRT THE AIRPLANE IN EARTH AXES SYSTEM ......
C
      CALL VELXYZ (UABE,UAP,XBA,WAP,DAE)
C
C  RELATIVE VELOCITY OF BLOCK WRT THE RAILS IN EARTH AXES SYSTEM ......
C
      DO 30 I=1,3
 30   RVBE(I) = USBE(I) - UABE(I)
C
C  RELATIVE VELOCITY OF BLOCK WRT RAILS IN RAIL AXES SYSTEM ......
C
      CALL MATMPY (RVBR,DER,RVBE,3,3,1)
C
C  FORCES ON SEAT IN RAIL AXES DUE TO RAIL RIGIDITY AND DAMPING ......
C
      FDEFL(1) = -SPR(1) * XBR(1) - DPG(1) * RVBR(1)
      FDEFL(2) = -SPR(2) * XBR(2) - DPG(2) * RVBR(2)
      FRVEL = SIGN(AMIN1(ABS(RVBR(3)),1.0),RVBR(3))
      FDEFL(3)= -FRICT*SQRT(FDEFL(1)**2+FDEFL(2)**2)*FRVEL
C
C  FORCES ON SEAT IN SEAT AXIS SYSTEM ......
C
      CALL MATMPY (FSEAT,DRS,FDEFL,3,3,1)
C
C  FORCES ON AIRPLANE IN AIRPLANE AXIS SYSTEM ......
```

```
C
      DO 40 I=1,3
 40   FDEFL(I)=-FDEFL(I)
      CALL MATMPY (FAIRP,DRA,FDEFL,3,3,1)
C
C  AIRPLANE MOMENT ARM ......
C
      CALL VECXYZ (ARM,TEMP,XRAIL,DRA,2)
C
C  MOMENTS ON SEAT ......
C
      CALL CRSPRD (TSEAT,XBS,FSEAT)
C
C  MOMENTS ON AIRPLANE ......
C
      CALL CRSPRD (TAIRP,ARM,FAIRP)
C
 50   RETURN
      END
```

```
      SUBROUTINE BRIDL2 (FAP,
     *                       APX,XPCDO,PT1,PT2)
C
C  THIS ROUTINE CALCULATES THE FORCE APPLICATION POINT OF A FORCE
C  APPLIED TO A TWO STRAND BRIDLE.
C
C  *****  BRIDL2 OUTPUTS  *****
C
C      FAP(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C               OF THE FORCE APPLICATION POINT (FT)
C
C  *****  BRIDL2 INPUTS  *****
C
C      APX(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C               OF THE BRIDLE APX (FT)
C      XPCDO(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C                 OF THE PARACHUTE (FT)
C      PT1(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C               OF BRIDLE ATTACHMENT POINT ONE (FT)
C      PT2(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C               OF BRIDLE ATTACHMENT POINT TWO (FT)
C
      DIMENSION FAP(3),APX(3),PT1(3),PT2(3),XPCDO(3)
C
      DIMENSION DELA1(3),DELA2(3),DCN(3),XI(3),DC3I(3),XIN(3),
     *          APXT(3),DIFF(3),UV1(3)
C
C  CALCULATE THE DIRECTION COSINES OF THE NORMAL TO VECTORS APX,PT1
C  AND APX,PT2 ......
C
      DO 10 I=1,3
      DELA1(I) = PT1(I) - APX(I)
 10   DELA2(I) = PT2(I) - APX(I)
C
      CALL CRSPRD (DCN,DELA1,DELA2)
C
      RN = SQRT (DCN(1)**2 + DCN(2)**2 + DCN(3)**2 )
      DO 20 I=1,3
 20   DCN(I) = DCN(I)/RN
C
C  CALCULATE THE NORMAL FROM APX TO VECTOR PT1,PT2 ......
C
      CALL LINDST (XI,RAI,DC3I,PT1,PT2,APX)
C
C  CALCULATE THE UNIT VECTOR FROM XPCDO TO PT1 ......
C
      DO 30 I=1,3
 30   DIFF(I) = PT1(I) - XPCDO(I)
C
      RESULT = SQRT(DIFF(1)**2+DIFF(2)**2+DIFF(3)**2)
C
      DO 40 I=1,3
 40   UV1(I) = DIFF(I)/RESULT
C
C
C  CALCULATE THE LOCATION OF THE BRIDLE CONFLUENCE POINT ......
C
      PHI = ARTAN2 ( DCN(1)*UV1(1)+DCN(2)*UV1(2)+DCN(3)*UV1(3),
```

```
     .              -(DC3I(1)*UV1(1)+DC3I(2)*UV1(2)+DC3I(3)*UV1(3)))
      SINPHI = SIN(PHI)
      COSPHI = COS(PHI)
      DO 50 I=1,3
 50   APXT(I) = XI(I) + RAI * (-DC3I(I)*COSPHI + DCN(I)*SINPHI)
C
C  CALCULATE THE UNIT VECTOR AND MAGNITUDE OF THE NORMAL FROM
C  THE BRIDLE CONFLUENCE POINT TO VECTOR PT1,PT2 .....
C
      CALL LINDST (XI,RAI,DC3I,PT1,PT2,APXT)
C
C  CALCULATE THE UNIT VECTOR FROM XPCDO TO APXT ......
C
      DO 60 I=1,3
 60   DIFF(I) = APXT(I) - XPCDO(I)
C
      RESULT = SQRT(DIFF(1)**2+DIFF(2)**2+DIFF(3)**2)
C
      DO 70 I=1,3
 70   UV1(I) = DIFF(I)/RESULT
C
C  DOT THE PARACHUTE LINE UNIT VECTOR ONTO DC3I ......
C
      CALL DOTPRD (COSINE,DC3I,UV1,3)
C
C  DETERMINE THE MAGNITUDE OF THE VECTOR FROM THE CONFLUENCE POINT
C  TO VECTOR PT1,PT2 ALONG THE LINE FORCE UNIT VECTOR  ......
C
      IF(COSINE.NE.0.) RI = RAI/COSINE
C
C  CALCULATE THE INTERSECTION OF THE PARACHUTE LINE FORCE VECTOR
C  WITH VECTOR PT1,PT2 ......
C
      DO 80 I=1,3
 80   XIN(I) = APXT(I) + RI * UV1(I)
C
C  DETERMINE THE FORCE APPLICATION POINT ......
C
      TEST = (XIN(1)-PT1(1))*(PT2(1)-PT1(1))+(XIN(2)-PT1(2))*
     .        (PT2(2)-PT1(2))+(XIN(3)-PT1(3))*(PT2(3)-PT1(3))
      IF(TEST.LE.0) GO TO 120
C
      R1IN = SQRT((XIN(1)-PT1(1))**2+(XIN(2)-PT1(2))**2+
     .             (XIN(3)-PT1(3))**2)
      R12 = SQRT ((PT2(1)-PT1(1))**2 + (PT2(2)-PT1(2))**2 +
     .             (PT2(3)-PT1(3))**2 )
      IF(R12-R1IN.GE.0) GO TO 100
C
      DO 90 I=1,3
 90   FAP(I) = PT2(I)
      GO TO 140
C
 100  DO 110 I=1,3
 110  FAP(I) = XIN(I)
      GO TO 140
C
 120  DO 130 I=1,3
 130  FAP(I) = PT1(I)
```

```
C
 140  RETURN
      END
```

```
      SUBROUTINE BRIDL3 (FAP,
     .                  APX,UV,XPCDO,PT1,PT2,PT3,XI)
C
C  ROUTINE FOR COMPUTING THE FORCE APPLICATION POINT FOR A BRIDLE
C  WITH THREE FLEXIBLE LINES
C
C  *****  BRIDL3 OUTPUTS  *****
C
C     FAP(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C              OF THE FORCE APPLICATION POINT (FT)
C
C  *****  BRIDL3 INPUTS  *****
C
C     APX(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C              OF THE BRIDLE APEX (FT)
C     UV(3)  - PARACHUTE LINE UNIT VECTOR
C     XPCDO(3)  - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITON VECTOR
C                 OF THE PARACHUTE (FT)
C     PT1(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C              OF BRIDLE ATTACHMENT POINT ONE (FT)
C     PT2(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C              OF BRIDLE ATTACHMENT POINT TWO (FT)
C     PT3(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C              OF BRIDLE ATTACHMENT POINT THREE (FT)
C     XI(3)  - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C              OF THE INTERSECTION OF THE BRIDLE ATTACHMENT POINTS PLANE
C              WITH THE PARACHUTE LINE FORCE VECTOR (FT)
C
      DIMENSION FAP(3),APX(3),UV(3),PT1(3),PT2(3),PT3(3),XI(3),
     .          XPCDO(3)
      DIMENSION XN123(3),DC123(3),XNI23(3),DCI23(3),XN213(3),
     .          DC213(3),XNI13(3),DCI13(3),XN312(3),DC312(3),
     .          XNI12(3),DCI12(3)
C
C  COMPUTE THE INTERSECTION OF THE NORMAL FROM POINT 1 TO
C  VECTOR 2,3 ......
C
      CALL LINDST (XN123,0,DC123,PT2,PT3,PT1)
C
C  COMPUTE THE INTERSECTION OF THE NORMAL FROM THE FORCE-PLANE
C  INTERSECTION TO VECTOR 2,3 ......
C
      CALL LINDST (XNI23,0,DCI23,PT2,PT3,XI)
C
C  ------  TEST FOR COMPRESSION IN LINE 1  -----
C
      TEST = DC123(1)*DCI23(1) + DC123(2)*DCI23(2) +
     .       DC123(3)*DCI23(3)
      IF (TEST) 10,10,20
C
C  LINE 1 UNDER COMPRESSION -  COMPUTE THE FORCE APPLICATION
C  POINT LYING ON VECTOR 2,3 ......
C
 10   CALL BRIDL2 (FAP,APX,XPCDO,PT2,PT3)
      GO TO 60
C
C  COMPUTE THE NORMAL FROM POINT 2 TO VECTOR 1,3 ......
C
```

```
 20   CALL LINDST (XN213,0,DC213,PT1,PT3,PT2)
C
C  COMPUTE THE NORMAL FROM THE FORCE-PLANE INTERSECTION TO
C  VECTOR 1,3 ......
C
      CALL LINDST (XNI13,0,DCI13,PT1,PT3,XI)
C
C  ------  TEST FOR COMPRESSION IN LINE 2  ------
C
      TEST = DC213(1)*DCI13(1) + DC213(2)*DCI13(2) +
     .        DC213(3)*DCI13(3)
      IF(TEST) 30,30,40
C
C  LINE 2 UNDER COMPRESSION -  COMPUTE THE FORCE APPLICATION
C  POINT LYING ON VECTOR 1,3 ......
C
 30   CALL BRIDL2 (FAP,APX,XPCOO,PT1,PT3)
      GO TO 80
C
C  COMPUTE THE NORMAL FROM POINT 3 TO VECTOR 1,2 ......
C
 40   CALL LINDST (XN312,0,DC312,PT1,PT2,PT3)
C
C  COMPUTE THE NORMAL FROM THE FORCE-PLANE INTERSECTION TO
C  VECTOR 1,2 ......
C
      CALL LINDST (XNI12,0,DCI12,PT1,PT2,XI)
C
C  -------  TEST FOR COMPRESSION IN LINE 3  ------
C
      TEST = DC312(1)*DCI12(1) + DC312(2)*DCI12(2) +
     .        DC312(3)*DCI12(3)
      IF (TEST) 50,50,60
C
C  LINE 3 UNDER COMPRESSION -  COMPUTE THE FORCE APPLICATION
C  POINT LYING ON VECTOR 1,2 ......
C
 50   CALL BRIDL2 (FAP,APX,XPCOO,PT1,PT2)
      GO TO 80
C
C  -------  ALL THREE LINES IN TENSION  ------
C
 60   DO 70 I=1,3
 70   FAP(I) = XI(I)
C
 80   RETURN
      END
```

```
      SUBROUTINE BRIDL4 (FAP,
     *                   APX,UV,XPCDO,AP1,AP2,AP3,AP4,XI)
C
C  THIS ROUTINE DETERMINES THE THREE BRIDLE ATTACHMENT POINTS
C  TO BE USED IN BRIDL3
C
C  *****  BRIDL4 OUTPUTS  *****
C
C     FAP(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C              OF THE FORCE APPLICATION POINT (FT)
C
C  *****  BRIDL4 INPUTS  *****
C
C     APX(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C              OF THE BRIDLE APEX (FT)
C     UV(3)  - PARACHUTE LINE FORCE UNIT VECTOR
C     XPCDO(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C                OF THE PARACHUTE (FT)
C     AP1(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITON VECTOR
C              OF BRIDLE ATTACHMENT POINT ONE (FT)
C     AP2(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C              OF BRIDLE ATTACHMENT POINT TWO (FT)
C     AP3(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C              OF BRIDLE ATTACHMENT POINT THREE (FT)
C     AP4(3) - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C              OF BRIDLE ATTACHMENT POINT FOUR (FT)
C     XI(3)  - X,Y,Z DECELERATED OBJECT BODY AXIS LINEAR POSITION VECTOR
C              OF THE INTERSECTION OF THE BRIDLE ATTACHMENT POINTS PLANE
C              WITH THE PARACHUTE LINE FORCE VECTOR (FT)
C
      DIMENSION FAP(3),APX(3),UV(3),AP1(3),AP2(3),AP3(3),AP4(4),
     *          XI(3),XPCDO(3)
      DIMENSION XN124(3),DC124(3),XNI24(3),DCI24(3)
C
C  THE FOUR ATTACHMENT POINTS OF THE BRIDLE ARE DESIGNATED AS 1,2,3,4
C  (NUMBERED CONSECUTIVELY IN A COUNTER CLOCKWISE DIRECTION)
C  LET POINTS 2 AND 4 DEFINE A LINE AND CHECK TO SEE WHICH SIDE OF
C  THE LINE THE FORCE-PLANE INTERSECTION LIES ON
C
      CALL LINDST (XN124,DC124,AP2,AP4,AP1)
      CALL LINDST (XNI24,DCI24,AP2,AP4,XI)
C
      TEST = DC124(1)*DCI24(1)+DC124(2)*DCI24(2)+DC124(3)*DCI24(3)
      IF(TEST.GE.0) GO TO 10
C
      CALL BRIDL3 (FAP,APX,UV,XPCDO,AP2,AP3,AP4,XI)
      GO TO 20
C
 10   CALL BRIDL3 (FAP,APX,UV,XPCDO,AP1,AP2,AP4,XI)
C
 20   RETURN
      END
```

```
      SUBROUTINE CAD (CF,EF,EFDOT,IEF,EL,ELDOT,IEL,WK,WKDOT,IWK,
     .                WB,WBDOT,IWB,
     .                FL,TCP,TIME,CEX,CSK,CI,C,VI,PA,TF,CVH,CBP,C1,
     .                CV,C2,TI,THA,B,BXP,PT,R,TYPE,
     .                TSO,FSO,TDE)
C
C  COMPUTES THE PERFORMANCE OF A CLOSED TELESCOPING TUBE
C  ACTING AGAINST A LOAD IN ANY G ENVIRONMENT AND USING A BURNING
C  PROPELLANT AS A SOURCE OF ENERGY ......
C
      DIMENSION TCP(5)
      COMMON / COVRLY / INST
      COMMON / CIO / IREAD,IWRITE,IDIAG
      DATA PIO2 / 1.57080 /
C
C  PRINT CATAPULT IGNITION STATEMENT ......
C
      IF(FL.NE.0) GO TO 20
      IF(INST.EQ.26) WRITE(6,10) TYPE,TIME
 10   FORMAT(/5X,A6,* IGNITION AT TIME = *,F10.4,* SEC*/)
      FL = 1.0
C
C  CALCULATE THE CATAPULT FORCE DECAY AFTER STRIPOFF ......
C
 20   IF(FL.EQ.1.) GO TO 40
      TASO = TIME - TSO
      IF(TASO.LT.TDE) GO TO 30
      FL = 3.
      CF = 0
      GO TO 150
 30   IF(TDE.NE.0) CF = FSO * COS(TASO/TDE * PIO2)
      GO TO 150
C
C  COMPUTE PROPELLANT CONSUMMED ......
C
 40   NA=TCP(2)
      W = CI + TBLU1 (WB,TCP(4),TCP(NA+4),1,-NA)
C
C  HAS ALL THE PROPELLANT BURNED ......
C
      IF(C.GE.W) GO TO 50
C
C  ALL BURNED ........
C
      W = C
C
C  COMPUTE INTERNAL VOLUME ......
C
 50   VOL = VI + PA * CEX * 12.
C
C  DON'T LET THE VOLUME DECREASE BELOW INITIAL VALUE ......
C
      IF(VOL.GE.VI) GO TO 60
C
      VOL = VI
C
 60   IF(W.NE.0.0) GO TO 70
      TEMP = TF
```

```
      GO TO 60
C
 70   TEMP = TF - (WK + EF + EL)/(W * CVH)
C
C  COMPUTE CHAMBER PRESSURE USING EQUATION OF STATE ......
C
 60   PRESS = 12.0 * R * TEMP * W / VOL
C
C  PRINT CATAPULT BURST STATEMENT (IF REQUIRED) ......
C
      IF(CBP.GE.PRESS) GO TO 100
      IF(INST.EQ.26) WRITE(6,90) TYPE,PRESS,TIME
 90   FORMAT(//5X,*GRRRRBBBBOOOOMMMM %/<  POOF HSSSSSSSS*,//5X,A8,
     . * BURST AT *,F10.4,* LBS PRESSURE AT TIME = *,F10.4,* SEC*//)
      STOP
C
C  HAS THE PRESSURE UNLOCKED THE PISTON YET ......
C
 100  IF(PRESS.GT.PT) GO TO 110
C
C  STILL LOCKED - SET CATAPULT FORCE TO ZERO
C
      CF = 0.0
      GO TO 120
C
C  UNLOCKED - HIT 'EM AND MOVE 'EM OUT ......
C
 110  CF = PA*PRESS*(1.-C1)
C
C  ******************************************************************
C  *                                                                *
C  *     COMPUTE INTERNAL FRICTIONAL ENERGY RATE, HEAT LOSS RATE,   *
C  *     CATAPULT WORK RATE, AND THE PROPELLANT BURN RATE           *
C  *                                                                *
C  ******************************************************************
C
C
C  COMPUTE THE INTERNAL FRICTIONAL ENERGY RATE (POWER) ......
C
 120  IF(IEF.NE.0) EFDOT = ABS (C1*PRESS*CV)
C
C  COMPUTE THE HEAT LOSS RATE ......
C
      IF(IEL.NE.0) ELDOT = ABS(C2*(TEMP - TI)*THA)
C
C  COMPUTE CATAPULT WORK RATE ......
C
      IF(IWK.NE.0) WKDOT = ABS(CF*CV)
C
C  COMPUTE PROPELLANT BURN RATE ......
C
      PBR = 0.0
      IF(W.GE.C) GO TO 130
      PBR = B*ABS(PRESS)**BXP
 130  IF(IWB.NE.0) WBDOT = PBR
C
C  ////////////////////////////////////////////////////////////////
C
```

```
C  WHEN STRIPOFF OCCURS ......
C
      IF(CEX.LT.CSK) GO TO 150
      FL = 2.
      EFDOT = 0.
      ELDOT = 0.
      WKDOT = 0.
      WBDOT = 0.
      TSO = TIME
      FSO = CF
C
C  PRINT CATAPULT STRIPOFF STATEMENT
C
      IF(INST.EQ.26) WRITE(6,140) TYPE,TIME
 140  FORMAT(/5X,A8,* STRIPOFF AT TIME = *,F10.4,* SEC*/)
C
 150  RETURN
      END
```

```
      SUBROUTINE CE (UCP,UDCP,IUCP,XCP,XDCP,IXCP,WCP,WDCP,IWCP,
     .               ECP,EDCP,IECP,SCD,SCDDOT,ISCD,SC,SCDOT,ISC,
     .               GX,GY,GZ,DR,FAD,TAD,WT,S,B,C,CIN,CX,CY,CZ,
     .               CL,CM,CN,ALPHA,BETA,VMACH,Q,ALT,SEP,
     .               SW,PC,CEW,CMI,CPI,CLP,CMQ,CNR,XSP,FAB,TAB,FDO,
     .               TDO,FAU,TAU,TRM)
C
C  ***************  CE OUTPUTS  ***************
C
C  LINEAR VELOCITIES - BODY AXIS
C
C     UCP(3)  - X,Y,Z LINEAR VELOCITY VECTOR OF THE CREWPERSON (FT/SEC)
C     UDCP(3) - X,Y,Z LINEAR VELOCITY RATE VECTOR OF THE CREWPERSON
C               (FT/SEC/SEC)
C     IUCP(3) - INTEGRATION CONTROL
C
C  LINEAR POSITIONS - EARTH SYSTEM
C
C     XCP(3)  - X,Y,Z LINEAR POSITION VECTOR OF THE CREWPERSON (FT)
C     XDCP(3) - X,Y,Z LINEAR POSITION RATE VECTOR OF THE CREWPERSON (FT/SEC)
C     IXCP(3) - INTEGRATION CONTROL
C
C  ANGULAR VELOCITIES - BODY AXIS
C
C     WCP(3)  - X,Y,Z ANGULAR VELOCITY COMPONENTS - P,Q,R  (DEG/SEC)
C     WDCP(3) - X,Y,Z ANGULAR VELOCITY RATE COMPONENTS (DEG/SEC/SEC)
C     IWCP(3) - INTEGRATION CONTROL
C
C  EULER ANGLES  --  EARTH TO BODY  --  YAW,PITCH,ROLL
C
C     ECP(3)  - EARTH TO CREWPERSON EULER ANGLES (DEG)
C     EDCP(3) - EULER ANGLE RATES (DEG/SEC)
C     IECP(3) - INTEGRATION CONTROL
C
C  SPINAL COMPRESSION VELOCITY ......
C
C     SCD      -  SPINAL COMPRESSION VELOCITY (FT/SEC)
C     SCDDOT   -  SPINAL COMPRESSION VELOCITY RATE (FT/SEC/SEC)
C     ISCD     -  INTEGRATION CONTROL
C
C  SPINAL COMPRESSION ......
C
C     SC      - SPINAL COMPRESSION (FT)
C     SCDOT   - SPINAL COMPRESSION RATE (FT/SEC)
C     ISC     - INTEGRATION CONTROL
C
C     GX      - CREWPERSON SYSTEM X-AXIS LOAD FACTOR (G)
C     GY      - CREWPERSON SYSTEM Y-AXIS LOAD FACTOR (G)
C     GZ      - CREWPERSON SYSTEM Z-AXIS LOAD FACTOR (G)
C     DR      - DYNAMIC RESPONSE
C
C     FAD(3)  - X,Y,Z BODY AXIS FORCE COMPONENTS OF THE AERODYNAMIC
C               FORCE ACTING ON THE CREWPERSON (LB)
C     TAD(3)  - X,Y,Z BODY AXIS TORQUE COMPONENTS OF THE AERODYNAMIC
C               TORQUE ACTING ON THE CREWPERSON (FT-LB)
C     WT      - WEIGHT OF THE CREWPERSON CORRESPONDING TO HIS
C               PERCENTILE PLUS CLOTHING AND EQUIPMENT (LB)
C     S       - AERODYNAMIC REFERENCE AREA (FT**2)
```

```
C     B          - AERODYNAMIC LATERAL REFERENCE LENGTH (FT)
C     C          - AERODYNAMIC LONGITUDINAL REFERENCE LENGTH (FT)
C     CIN(4)     - CREWPERSON INERTIA PROPERTIES TO BE USED AFTER
C                  SEAT/CREWPERSON SEPARATION
C                          CIN(1) = IXX
C                          CIN(2) = IYY
C                          CIN(3) = IZZ
C                          CIN(4) = IXZ
C     CX         - X AXIS AERODYNAMIC FORCE COEFFICIENT
C     CY         - Y AXIS AERODYNAMIC FORCE COEFFICIENT
C     CZ         - Z AXIS AERODYNAMIC FORCE COEFFICIENT
C     CL         - AERODYNAMIC ROLLING MOMENT COEFFICIENT
C     CM         - AERODYNAMIC PITCHING MOMENT COEFFICIENT
C     CN         - AERODYNAMIC YAWING MOMENT COEFFICIENT
C     ALPHA      - CREWPERSON ANGLE OF ATTACK (DEG)
C     BETA       - CREWPERSON SIDESLIP ANGLE (DEG)
C     VMACH      - CREWPERSON MACH NUMBER
C     Q          - CREWPERSON DYNAMIC PRESSURE (LB/FT**2)
C     ALT        - CREWPERSON ALTITUDE (FT)
C     SEP        - SEAT/CREWPERSON SEPARATION FLAG FOR OUTPUT
C                  (1 = SEPARATION)
C
C  ***************  CE INPUTS  ***************
C
C     SW       - FLAG FOR SEAT/CREWPERSON SEPARATION
C                (1 = SEPARATION)
C     PC       - CREWPERSON PERCENTILE
C     CEW      - WEIGHT OF CREWPERSON CLOTHING AND EQUIPMENT (LB)
C     CMI(3) - CREWPERSON MOMENT OF INERTIA VECTOR - IXX,IYY,IZZ
C                (SLUG-FT**2)
C     CPI(3) - CREWPERSON PRODUCT OF INERTIA VECTOR - IXY,IXZ,IYZ
C                (SLUG-FT**2)
C     CLP      - AERODYNAMIC ROLL DAMPING COEFFICIENT (1/DEG)
C     CMQ      - AERODYNAMIC PITCH DAMPING COEFFICIENT (1/DEG)
C     CNR      - AERODYNAMIC YAW DAMPING COEFFICIENT (1/DEG)
C     XSP(3) - X,Y,Z CREWPERSON SYSTEM POSITION VECTOR OF THE BASE
C                OF THE SPINE (FT)
C     FAB(3) - X,Y,Z BODY AXIS FORCE COMPONENTS ACTING ON THE CREWPERSON
C                FROM THE RESTRAINT COMPONENT (LB)
C     TAB(3) - X,Y,Z BODY AXIS TORQUE COMPONENTS ACTING ON THE CREWPERSON
C                FROM THE RESTAINT COMPONENT (FT-LB)
C     FDO(3) - X,Y,Z BODY AXIS FORCE COMPONENTS ACTING ON THE CREWPERSON
C                FROM THE PARACHUTE LINE COMPONENT (LB)
C     TDO(3) - X,Y,Z BODY AXIS TORQUE COMPONENTS ACTING ON THE CREWPERSON
C                FROM THE PARACHUTE LINE COMPONENT (LB)
C     FAU(3) - X,Y,Z BODY AXIS FORCE COMPONENTS ACTING ON THE CREWPERSON
C                --- AN AUXILIARY INPUT ---    (LB)
C     TAU(3) - X,Y,Z BODY AXIS TORQUE COMPONENTS ACTING ON THE CREWPERSON
C                --- AN AUXILIARY INPUT ---    (FT-LB)
C     TRM(3) - X,Y,Z PARENT BODY EARTH VELOCITY COMPONENTS
C                TO DETERMINE POSITION RATES DURING TRIM (FT/SEC)
C
C  DIMENSIONS OF CALLING ARGUMENTS ......
C
      DIMENSION UCP(3),UDCP(3),IUCP(3),XCP(3),XDCP(3),IXCP(3),
     .          WCP(3),WDCP(3),IWCP(3),ECP(3),EDCP(3),IECP(3),
     .          FAD(3),TAD(3),CIN(4),
     .          CMI(3),CPI(3),XSP(3),FAB(3),TAB(3),FDO(3),TDO(3),
```

```
     .           FAU(3),TAU(3),TRM(3)
C
C  INTERNAL DIMENSIONS ......
C
      DIMENSION TINER(3,3),TEMP1(3),TEMP2(3),TEMP3(3),
     .           UWB(3),UW(3),UO(3),ECPIR(3),WCPIR(3),EDCPIR(3),
     .           WDCPIR(3),DEC(3,3),DCE(3,3),TBLCP(10),TBLCPWT(10),
     .           TBLIXX(10),TBLIYY(10),TBLIZZ(10),TBLIXZ(10),TBLS(10),
     .           TBLB(10),TBLC(10),F(3),T(3)
C
      COMMON / CICCAL / ICCAL
      COMMON / COVRLY / INST
      COMMON / CTIME / TIME
      COMMON / CSSFLG / SSFLG
      COMMON / CIO / IREAD,IWRITE,IDIAG
C
      DATA RPD,DPR / .01745329,57.29578 /
      DATA GRAV /32.174/
C
      DATA TBLCP /5.,15.,25.,35.,45.,55.,65.,75.,85.,95./
      DATA TBLCPWT /132.3,142.7,149.1,154.3,159.3,164.6,170.5,
     .              177.4,186.5,200.9/
      DATA TBLIXX /10.53,11.51,12.10,12.56,13.00,13.47,14.00,
     .             14.64,15.51,16.97/
      DATA TBLIYY /10.36,11.57,12.16,12.61,13.04,13.50,14.01,
     .             14.63,15.48,16.92/
      DATA TBLIZZ /1.68,1.78,1.85,1.90,1.95,2.01,2.07,2.14,
     .             2.24,2.41/
      DATA TBLIXZ /-.52,-.51,-.50,-.50,-.49,-.48,-.48,-.47,-.47,-.46/
      DATA TBLS /7.46,7.65,8.09,8.30,8.49,8.67,8.87,9.10,9.38,9.85/
      DATA TBLB /1.38,1.41,1.44,1.46,1.48,1.50,1.52,1.54,1.57,1.62/
      DATA TBLC /5.43,5.55,5.63,5.69,5.74,5.79,5.84,5.89,5.97,6.10/
C
C  ***************************
C  *****  INITIALIZATION  *****
C  ***************************
C
      IF(ICCAL.NE.1) GO TO 20
C
      CX = CY = CZ = CL = CM = CN = ALPHA = BETA = VMACH = 0
      Q = SEP = 0
      DO 10 I=1,3
      TRM(I) = FAD(I) = TAD(I) = 0
      IF(XSP(I) .EQ. 0.99999) XSP(I) = 0
      IF(FAB(I) .EQ. 0.99999) FAB(I) = 0
      IF(TAB(I) .EQ. 0.99999) TAB(I) = 0
      IF(FDO(I) .EQ. 0.99999) FDO(I) = 0
      IF(TDO(I) .EQ. 0.99999) TDO(I) = 0
      IF(FAU(I) .EQ. 0.99999) FAU(I) = 0
      IF(TAU(I) .EQ. 0.99999) TAU(I) = 0
 10   CONTINUE
      IF(SW .EQ. 0.99999) SW = 0
      WT     = TBLU1(PC,TBLCP,TBLCPWT,1,-10) + CEW
      S      = TBLU1(PC,TBLCP,TBLS,1,-10)
      B      = TBLU1(PC,TBLCP,TBLB,1,-10)
      C      = TBLU1(PC,TBLCP,TBLC,1,-10)
C
C  *****  CALCULATE THE CREWPERSON INERTIAS FOR USE AFTER  *****
```

```
C                         SEAT/CREWPERSON SEPARTION
C
      CIN(1) = TBLU1(PC,TBLCP,TBLIXX,1,-10)
      CIN(2) = TBLU1(PC,TBLCP,TBLIYY,1,-10)
      CIN(3) = TBLU1(PC,TBLCP,TBLIZZ,1,-10)
      CIN(4) = TBLU1(PC,TBLCP,TBLIXZ,1,-10)
C
C  ////////////////////////////////////////////////////////////////
C
C  CHANGE FROM DEGREES TO RADIANS ......
C
 20   DO 30 I=1,3
      ECPIR(I) = ECP(I) * RPD
 30   WCPIR(I) = WCP(I) * RPD
C
C  COMPUTE THE DIRECTION COSINE MATRICES ......
C
      CALL DIRCOS (DEC,ECPIR)
      CALL TRANS (DCE,DEC,3,3)
C
C  ESTABLISH POSITIVE ALTITUDE ......
C
      ALT = - XCP(3)
C
C  BYPASS THE AERODYNAMIC CALCULATIONS UP TO SEAT/CREWPERSON
C  SEPARATION ......
C
      IF(SW.EQ.1.) GO TO 40
C
C  SET UP THE SEATED CREWPERSON INERTIA TENSOR ......
C
      TINER(1,1) =  CMI(1)
      TINER(1,2) = -CPI(1)
      TINER(1,3) = -CPI(2)
      TINER(2,1) = -CPI(1)
      TINER(2,2) =  CMI(2)
      TINER(2,3) = -CPI(3)
      TINER(3,1) = -CPI(2)
      TINER(3,2) = -CPI(3)
      TINER(3,3) =  CMI(3)
      GO TO 110
C
C  SET UP THE EXTENDED CREWPERSON INERTIA TENSOR ......
C
 40   TINER(1,1) = CIN(1)
      TINER(1,2) = 0
      TINER(1,3) = -CIN(4)
      TINER(2,1) = 0
      TINER(2,2) = CIN(2)
      TINER(2,3) = 0
      TINER(3,1) = -CIN(4)
      TINER(3,2) = 0
      TINER(3,3) = CIN(3)
C
C  WRITE THE SEAT/CREWPERSON SEPARATION MESSAGE ......
C
      IF(SEP.EQ.1.) GO TO 60
      SEP = 1.
```

```
      IF(INST.EQ.26) WRITE(6,50) TIME
 50   FORMAT(/5X,*SEAT/CREWPERSON SEPARATION AT TIME = *,F10.4,* SEC*/)
C
C  OBTAIN SPEED OF SOUND, AIR DENSITY, AND WIND VELOCITY ......
C
 60   CALL ATMOS (VS,RHO,ALT,UW,0,0,0)
C
C  PUT THE WIND INTO BODY AXIS ......
C
      CALL MATMPY (UWB,DEC,UW,3,3,1)
C
C  ADD THE WIND VELOCITY TO THE CREWPERSON VELOCITY ......
C
      DO 90 I=1,3
 90   UO(I) = UCP(I) - UWB(I)
C
C  CALCULATE THE AERO VARIABLES ......
C
      IF(UO(1).EQ.0. .AND. UO(3).EQ.0.) UO(1) = .01
      ALPHA = ARTAN2(UO(3),UO(1)) * DPR
      CALL DOTPRD (VBAR2,UO,UO,3)
      VBAR = SQRT(VBAR2)
      BETA = ASIN(UO(2)/VBAR) * DPR
      VMACH = VBAR/VS
C
C  COMPUTE DYNAMIC PRESSURE X REFERENCE AREA ......
C
      Q = .5 * RHO * VBAR2
      QAC = Q * S
C
C  CALCULATE THE AERODYNAMIC COEFFICIENTS ......
C
      TBLALPH = ALPHA
      IF(ALPHA. LT. 0) TBLALPH = ALPHA + 360.0
      TBLBETA = ABS(BETA)
      CALL CEAERO (CX,CY,CZ,CL,CM,CN,TBLALPH,TBLBETA,PC)
C
      CY = CY * SIGN(1.,BETA)
      CL = -CL * SIGN(1.,BETA)
      CN = -CN * SIGN(1.,BETA)
C
C  ADD DAMPING TERMS FOR AN AIRSPEED GREATER THAT .1 FT/SEC ......
C
      IF(VBAR .LE. 0.1) GO TO 100
C
      CO2V = C/(VBAR+VBAR)
      BO2V = B/(VBAR+VBAR)
C
C  ADD ROLL DAMPING ......
C
      CL = CL + CLP * WCP(1) * BO2V
C
C  ADD PITCH DAMPING ......
C
      CM = CM + CMQ * WCP(2) * CO2V
C
C  ADD YAW DAMPING ......
C
```

```
      CN = CN + CNR * WCP(3) * BO2V
C
C  **********  AERODYNAMIC FORCES  **********
C
 100  FAD(1) = QAC * CX
      FAD(2) = QAC * CY
      FAD(3) = QAC * CZ
C
C  **********  AERODYNAMIC TORQUES  **********
C
      TAD(1) = QAC * B * CL
      TAD(2) = QAC * C * CM
      TAD(3) = QAC * B * CN
C
C  *****  SUM FORCES (INCLUDING GRAVITY) AND MOMENTS  *****
C
 110  DO 120 I=1,3
      F(I) = FAB(I) + FDO(I) + FAU(I) + FAD(I) + WT * DEC(I,3)
     .       * SSFLG
      T(I) = TAB(I) + TDO(I) + TAU(I) + TAD(I)
 120  CONTINUE
C
C  CALCULATE THE DYNAMIC RESPONSE ......
C
      DR = SC * 86.977
C
C  ************************************
C  *****  ANGULAR VELOCITY EQUATIONS  *****
C  ************************************
C
C  CALCULATE TINER X WCPIR ......
C
      CALL MATMPY (TEMP1,TINER,WCPIR,3,3,1)
C
C  CALCULATE WCPIR X (TINER * WCPIR) ......
C
      CALL CRSPRD (TEMP2,WCPIR,TEMP1)
C
C  SUM TERMS TO OBTAIN TOTAL TORQUE ......
C
      DO 130 I=1,3
 130  TEMP3(I) = T(I) - TEMP2(I)
C
C  CALCULATE WDCPIR ......
C
      CALL LUEQS (TINER,TEMP1,TEMP3,TEMP2,3,1,3,3,3,1.E-14,IERROR)
      IF(IERROR.NE.1) GO TO 150
      WRITE(6,140)
 140  FORMAT(* INERTIA MATRIX OF CREWPERSON IS SINGULAR...RUN STOPPED*)
      STOP
C
 150  DO 160 I=1,3
 160  IF(IWCP(I).NE.0) WDCPIR(I) = TEMP1(I)
C
C  ********************************
C  *****  EULER ANGLE EQUATIONS  *****
C  ********************************
C
```

```
      CALL EARATE (TEMP1,WCPIR,ECPIR)
      DO 170 I=1,3
 170  IF(IECP(I).NE.0) EDCPIR(I) = TEMP1(I)
C
C  ***************************************
C  *****  LINEAR VELOCITY EQUATIONS  *****
C  ***************************************
C
C  CALCULATE THE CORIOLIS ACCELERATION (WCPIR X UCP) ......
C
      CALL CRSPRD (TEMP1,WCPIR,UCP)
C
C  CALCULATE F/M ......
C
      CPMASS = WT/GRAV
      DO 180 I=1,3
 180  TEMP2(I) = F(I)/CPMASS
C
C  SUM THE ACCELERATION COMPONENTS ......
C
      DO 190 I=1,3
 190  IF(IUCP(I).NE.0) UDCP(I) = TEMP2(I) - TEMP1(I)
C
C  =====  CALCULATE THE LOAD FACTORS  =====
C
C  DETERMINE WDCPIR X XSP ......
C
      CALL CRSPRD (TEMP1,WDCPIR,XSP)
C
C  DETERMINE WCPIR X (WCPIR X XSP) ......
C
      CALL CRSPRD (TEMP2,WCPIR,XSP)
      CALL CRSPRD (TEMP3,WCPIR,TEMP2)
C
C  DETERMINE THE LOAD FACTORS ......
C
      GX = -(F(1)/CPMASS + TEMP1(1) + TEMP3(1))/GRAV
      GY = -(F(2)/CPMASS + TEMP1(2) + TEMP3(2))/GRAV
      GZ = -(F(3)/CPMASS + TEMP1(3) + TEMP3(3))/GRAV
C
C  ***************************************
C  *****  LINEAR POSITION EQUATIONS  *****
C  ***************************************
C
      CALL MATMPY (TEMP1,DCE,UCP,3,3,1)
      DO 200 I=1,3
 200  IF(IXCP(I).NE.0) XDCP(I) = TEMP1(I)
C
C  ******************************************
C  *****  SPINAL COMPRESSION EQUATIONS  *****
C  ******************************************
C
C  SPINAL COMPRESSION VELOCITY EQUATION ......
C
      IF(ISCD.NE.0) SCDDOT = -23.6992 * SCD - 2798.41 * SC
     .                       + GRAV * GZ
C
C  SPINAL COMPRESSION EQUATION ......
```

```
C
      IF(ISC.NE.0) SCDOT = SCD
C
C  DURING TRIM, SUBTRACT TRIM VELOCITY FROM POSITION RATES ......
C
      IF(INST.NE.31) GO TO 220
      DO 210 I=1,3
 210  IF(IXCP(I).NE.0) XDCP(I) = XDCP(I) - TRM(I)
C
C  *****  CHANGE FROM RADIANS TO DEGREES  *****
C
 220  DO 230 I=1,3
      EDCP(I) = EDCPIR(I) * DPR
 230  WDCP(I) = WDCPIR(I) * DPR
C
      RETURN
      END
```

```
      SUBROUTINE CEAERO (CX,CY,CZ,CL,CM,CN,ALPHA,BETA,PC)
      DIMENSION TCX(8,13,2),TCY(8,13,2),TCZ(8,13,2),
     .          TCL(8,13,2),TCM(8,13,2),TCN(8,13,2),
     .          TBETA(8),TALPHA(13),TPC(2)
C
      DATA (((TCX(I,J,K), I=1,8), J=1,13), K=1,1) /
     *-0.7063,-0.7063,-0.6642,-0.6099,-0.5004,-0.3017,-0.0729,-0.0898,
     *-0.6176,-0.6176,-0.6182,-0.5560,-0.4141,-0.2508,-0.0961,-0.0898,
     *-0.3174,-0.3174,-0.3244,-0.3173,-0.2471,-0.1553,-0.1235,-0.0898,
     * 0.0191, 0.0191, 0.0617, 0.0879,-0.0199,-0.0454,-0.0534,-0.0898,
     * 0.2640, 0.2640, 0.3473, 0.2583, 0.2129, 0.0899,-0.0035,-0.0898,
     * 0.4882, 0.4882, 0.5048, 0.4907, 0.3730, 0.1918, 0.0516,-0.0898,
     * 0.5864, 0.5864, 0.6006, 0.5669, 0.4485, 0.2593, 0.0663,-0.0898,
     * 0.5250, 0.5250, 0.5167, 0.5363, 0.4208, 0.2283, 0.0359,-0.0898,
     * 0.3631, 0.3631, 0.2653, 0.2458, 0.1589, 0.0462,-0.0635,-0.0898,
     * 0.0378, 0.0378, 0.0303, 0.0113,-0.0147,-0.0384,-0.0424,-0.0898,
     *-0.3097,-0.3097,-0.3329,-0.3566,-0.2923,-0.1970,-0.0743,-0.0898,
     *-0.6183,-0.6183,-0.6155,-0.5990,-0.5208,-0.3282,-0.0960,-0.0898,
     *-0.7063,-0.7063,-0.6642,-0.6099,-0.5004,-0.3017,-0.0729,-0.0898/
C
      DATA (((TCX(I,J,K), I=1,8), J=1,13), K=2,2) /
     *-0.7327,-0.7327,-0.6897,-0.6126,-0.4824,-0.3180,-0.0855,-0.0376,
     *-0.6037,-0.6037,-0.5943,-0.5071,-0.3853,-0.2623,-0.0980,-0.0376,
     *-0.2935,-0.2935,-0.3034,-0.2810,-0.2507,-0.1197,-0.0730,-0.0376,
     * 0.0150, 0.0150, 0.0234, 0.0829, 0.0387, 0.0369,-0.0377,-0.0376,
     * 0.2640, 0.2640, 0.2916, 0.2494, 0.1968, 0.0811, 0.0115,-0.0376,
     * 0.5036, 0.5036, 0.5025, 0.4278, 0.3519, 0.2251, 0.1094,-0.0376,
     * 0.6700, 0.6700, 0.6197, 0.5528, 0.4423, 0.3026, 0.1535,-0.0376,
     * 0.5614, 0.5614, 0.5564, 0.5682, 0.4418, 0.2942, 0.1685,-0.0376,
     * 0.2693, 0.2693, 0.0530, 0.2388, 0.1663, 0.1055, 0.0356,-0.0376,
     * 0.0059, 0.0059, 0.0110,-0.0123,-0.0100,-0.0067,-0.0367,-0.0376,
     *-0.3350,-0.3350,-0.3443,-0.3122,-0.2478,-0.1431,-0.0369,-0.0376,
     *-0.6409,-0.6409,-0.6366,-0.5762,-0.4843,-0.2956,-0.2037,-0.0376,
     *-0.7327,-0.7327,-0.6897,-0.6126,-0.4824,-0.3180,-0.0855,-0.0376/
C
      DATA (((TCY(I,J,K), I=1,8), J=1,13), K=1,1) /
     * 0.    , 0.0559,-0.1501,-0.4278,-0.6545,-0.7659,-0.7094,-0.6854,
     * 0.    , 0.0273,-0.1408,-0.4061,-0.6172,-0.6803,-0.6828,-0.6854,
     * 0.    ,-0.0095,-0.1570,-0.3584,-0.5500,-0.6365,-0.6548,-0.6854,
     * 0.    ,-0.0230,-0.1689,-0.3046,-0.4497,-0.5521,-0.6162,-0.6854,
     * 0.    , 0.0061,-0.1448,-0.3696,-0.5458,-0.6565,-0.6287,-0.6854,
     * 0.    ,-0.0652,-0.2422,-0.4668,-0.6133,-0.6874,-0.6840,-0.6854,
     * 0.    , 0.0242,-0.2139,-0.5061,-0.6741,-0.7252,-0.7153,-0.6854,
     * 0.    ,-0.0401,-0.2845,-0.4625,-0.6174,-0.7046,-0.7335,-0.6854,
     * 0.    ,-0.0886,-0.2043,-0.4672,-0.6197,-0.7370,-0.7753,-0.6854,
     * 0.    ,-0.0842,-0.2098,-0.4341,-0.6554,-0.6032,-0.7623,-0.6854,
     * 0.    ,-0.0822,-0.2640,-0.4925,-0.7022,-0.7839,-0.7906,-0.6854,
     * 0.    ,-0.0080,-0.1922,-0.4524,-0.6568,-0.7583,-0.7586,-0.6854,
     * 0.    , 0.0559,-0.1501,-0.4278,-0.6545,-0.7659,-0.7094,-0.6854/
C
      DATA (((TCY(I,J,K), I=1,8), J=1,13), K=2,2) /
     * 0.    ,-0.0446,-0.1359,-0.3641,-0.5237,-0.6185,-0.5995,-0.5836,
     * 0.    ,-0.0828,-0.1411,-0.3290,-0.5048,-0.5731,-0.5610,-0.5836,
     * 0.    ,-0.0153,-0.1308,-0.2974,-0.4289,-0.5629,-0.5607,-0.5836,
     * 0.    ,-0.0293,-0.1588,-0.2686,-0.4045,-0.4979,-0.5535,-0.5836,
     * 0.    ,-0.0121,-0.1999,-0.3636,-0.4897,-0.5510,-0.5734,-0.5836,
     * 0.    ,-0.0678,-0.2762,-0.4458,-0.5490,-0.6332,-0.5848,-0.5836,
     * 0.    ,-0.0238,-0.2042,-0.4314,-0.5635,-0.6022,-0.6180,-0.5836,
```

```
* 0.     ,-0.0144,-0.2190,-0.3961,-0.5249,-0.6027,-0.6123,-0.5836,
* 0.     , 0.0006,-0.1797,-0.3591,-0.5151,-0.5870,-0.5874,-0.5836,
* 0.     ,-0.0260,-0.1211,-0.3306,-0.5298,-0.5897,-0.6420,-0.5836,
* 0.     ,-0.0164,-0.1769,-0.4301,-0.5683,-0.6462,-0.6588,-0.5836,
* 0.     , 0.0102,-0.1165,-0.3813,-0.5472,-0.6291,-0.6089,-0.5836,
* 0.     ,-0.0446,-0.1359,-0.3641,-0.5237,-0.6185,-0.5995,-0.5836/
C
  DATA (((TCZ(I,J,K), I=1,8), J=1,13), K=1,1) /
*-0.0230,-0.0230,-0.0177,-0.0375,-0.0600,-0.0599,-0.0627,-0.0422,
*-0.1042,-0.1042,-0.1011,-0.0918,-0.0895,-0.1192,-0.0741,-0.0422,
*-0.2552,-0.2552,-0.2611,-0.2222,-0.1591,-0.0801,-0.0605,-0.0422,
*-0.2772,-0.2772,-0.2784,-0.2566,-0.2339,-0.1772,-0.0735,-0.0422,
*-0.2844,-0.2844,-0.2632,-0.2797,-0.2245,-0.1709,-0.0846,-0.0422,
*-0.1581,-0.1581,-0.1676,-0.1522,-0.1390,-0.0891,-0.0466,-0.0422,
* 0.0156, 0.0156, 0.0191, 0.0100, 0.0033,-0.0322,-0.0610,-0.0422,
* 0.0796, 0.0796, 0.0715, 0.0576, 0.0419, 0.0071,-0.0247,-0.0422,
* 0.1269, 0.1269, 0.1873, 0.1424, 0.0976, 0.0237,-0.0435,-0.0422,
* 0.2862, 0.2862, 0.2628, 0.2000, 0.1228, 0.0192,-0.0618,-0.0423,
* 0.2002, 0.2002, 0.1883, 0.1504, 0.0733,-0.0083,-0.0988,-0.0422,
* 0.1124, 0.1124, 0.1093, 0.0612, 0.0088,-0.0330,-0.0720,-0.0422,
*-0.0230,-0.0230,-0.0177,-0.0375,-0.0600,-0.0599,-0.0627,-0.0422/
C
  DATA (((TCZ(I,J,K), I=1,8), J=1,13), K=2,2) /
* 0.0471, 0.0471, 0.0202,-0.0124,-0.0694,-0.0836,-0.0684,-0.0327,
*-0.0669,-0.0669,-0.0671,-0.0987,-0.0860,-0.1145,-0.0708,-0.0327,
*-0.2314,-0.2314,-0.2186,-0.1931,-0.1657,-0.1251,-0.0754,-0.0327,
*-0.2429,-0.2429,-0.2506,-0.2488,-0.2234,-0.1389,-0.1007,-0.0327,
*-0.2743,-0.2743,-0.2768,-0.2557,-0.2092,-0.1461,-0.0640,-0.0327,
*-0.1847,-0.1847,-0.1738,-0.2008,-0.1823,-0.1144,-0.0792,-0.0327,
*-0.0049,-0.0049,-0.0344,-0.0633,-0.0657,-0.0574,-0.0351,-0.0327,
* 0.1996, 0.1996, 0.1536, 0.1157, 0.0469, 0.0527, 0.0660, 0.0327,
* 0.1680, 0.1680, 0.1749, 0.1368, 0.0995, 0.0483,-0.0098,-0.0327,
* 0.2679, 0.2679, 0.2453, 0.2004, 0.1273, 0.0304, 0.0169,-0.0327,
* 0.1896, 0.1896, 0.1931, 0.1472, 0.0684, 0.0024,-0.0611,-0.0327,
* 0.0763, 0.0763, 0.0649, 0.0273,-0.0258,-0.0601,-0.0621,-0.0327,
* 0.0471, 0.0471, 0.0202,-0.0124,-0.0694,-0.0836,-0.0684,-0.0327/
C
  DATA (((TCL(I,J,K), I=1,8), J=1,13), K=1,1) /
* 0.     , 0.0051, 0.0171, 0.0343, 0.0272, 0.0189, 0.0021,-0.0025,
* 0.     , 0.0121, 0.0395, 0.0560, 0.0430, 0.0381, 0.0173,-0.0024,
* 0.     , 0.0007, 0.0172, 0.0372, 0.0396, 0.0412, 0.0339,-0.0024,
* 0.     ,-0.0016, 0.0070, 0.0204, 0.0467, 0.0540, 0.0405,-0.0024,
* 0.     , 0.0112, 0.0083, 0.0145, 0.0478, 0.0593, 0.0410,-0.0025,
* 0.     , 0.0024, 0.0037, 0.0224, 0.0311, 0.0417, 0.0367,-0.0025,
* 0.     ,-0.0012,-0.0012, 0.0070, 0.0178,-0.0151, 0.0029,-0.0024,
* 0.     ,-0.0123,-0.0186,-0.0125,-0.0044,-0.0139,-0.0064,-0.0024,
* 0.     , 0.0003,-0.0163,-0.0336,-0.0389,-0.0367,-0.0216,-0.0024,
* 0.     , 0.0034,-0.0066,-0.0272,-0.0409,-0.0327,-0.0184,-0.0024,
* 0.     , 0.0004,-0.0036,-0.0132,-0.0186,-0.0301,-0.0215,-0.0024,
* 0.     , 0.0061, 0.0120, 0.0159, 0.0081,-0.0022,-0.0127,-0.0024,
* 0.     , 0.0051, 0.0171, 0.0343, 0.0272, 0.0189, 0.0021,-0.0025/
C
  DATA (((TCL(I,J,K), I=1,8), J=1,13), K=2,2) /
* 0.     , 0.0184, 0.0292, 0.0310, 0.0275, 0.0158, 0.0066, 0.0080,
* 0.     , 0.0332, 0.0417, 0.0564, 0.0543, 0.0350, 0.0158, 0.0079,
* 0.     , 0.0136, 0.0229, 0.0339, 0.0440, 0.0498, 0.0336, 0.0079,
* 0.     , 0.0039, 0.0112, 0.0208, 0.0464, 0.0549, 0.0377, 0.0080,
* 0.     , 0.0043, 0.0099, 0.0242, 0.0473, 0.0537, 0.0309, 0.0080,
```

```
     * 0.    , 0.0132, 0.0030, 0.0134, 0.0267, 0.0390,-0.0171, 0.0080,
     * 0.    ,-0.0077,-0.0000,-0.0051,-0.0041, 0.0001,-0.0007, 0.0079,
     * 0.    ,-0.0006,-0.0084,-0.0113,-0.0079,-0.0045, 0.0127, 0.0079,
     * 0.    ,-0.0084,-0.0309,-0.0251,-0.0287,-0.0265,-0.0086, 0.0079,
     * 0.    ,-0.0021,-0.0123,-0.0268,-0.0267,-0.0215,-0.0097, 0.0080,
     * 0.    , 0.0032,-0.0003,-0.0099,-0.0151, 0.0169, 0.0032, 0.0075,
     * 0.    , 0.0137, 0.0156, 0.0133, 0.0059,-0.0024,-0.0069, 0.0080,
     * 0.    , 0.0184, 0.0292, 0.0310, 0.0275, 0.0158, 0.0068, 0.0080/
C
      DATA (((TCM(I,J,K), I=1,8), J=1,13), K=1,1) /
     * 0.0179, 0.0179, 0.0127, 0.0117, 0.0107, 0.0025,-0.0022,-0.0065,
     *-0.0291,-0.0291,-0.0114, 0.0018,-0.0017,-0.0023,-0.0033,-0.0065,
     *-0.0066,-0.0066, 0.0017, 0.0098, 0.0029,-0.0058,-0.0042,-0.0065,
     * 0.0149, 0.0149, 0.0119, 0.0082, 0.0169, 0.0058,-0.0009,-0.0065,
     * 0.0530, 0.0530, 0.0468, 0.0473, 0.0352, 0.0255, 0.0056,-0.0065,
     * 0.0578, 0.0578, 0.0558, 0.0515, 0.0433, 0.0301, 0.0097,-0.0065,
     * 0.0155, 0.0155, 0.0161, 0.0117, 0.0114,-0.0151,-0.0063,-0.0065,
     *-0.0578,-0.0578,-0.0471,-0.0470,-0.0216,-0.0203,-0.0085,-0.0065,
     *-0.0622,-0.0622,-0.0615,-0.0567,-0.0435,-0.0228,-0.0088,-0.0065,
     *-0.0224,-0.0224,-0.0185,-0.0098,-0.0129,-0.0066,-0.0048,-0.0065,
     * 0.0306, 0.0306, 0.0306, 0.0303, 0.0237, 0.0177, 0.0038,-0.0065,
     * 0.0438, 0.0438, 0.0408, 0.0355, 0.0254, 0.0185, 0.0109,-0.0065,
     * 0.0179, 0.0179, 0.0127, 0.0117, 0.0107, 0.0025,-0.0022,-0.0065/
C
      DATA (((TCM(I,J,K), I=1,8), J=1,13), K=2,2) /
     *-0.0234,-0.0234,-0.0191,-0.0135,-0.0097,-0.0029,-0.0008,-0.0004,
     *-0.0401,-0.0401,-0.0406,-0.0238,-0.0189,-0.0079, 0.0045,-0.0004,
     *-0.0232,-0.0232,-0.0076,-0.0120,-0.0092,-0.0017,-0.0011,-0.0004,
     * 0.0085, 0.0085, 0.0079, 0.0039, 0.0066, 0.0043, 0.0002,-0.0004,
     * 0.0425, 0.0425, 0.0444, 0.0421, 0.0351, 0.0192, 0.0028,-0.0004,
     * 0.0543, 0.0543, 0.0439, 0.0408, 0.0316, 0.0512,-0.0082,-0.0004,
     * 0.0249, 0.0249, 0.0199, 0.0165, 0.0107,-0.0049,-0.0056,-0.0004,
     *-0.0104,-0.0104,-0.0120,-0.0104,-0.0119,-0.0175,-0.0100,-0.0004,
     *-0.0441,-0.0441,-0.0415,-0.0269,-0.0193,-0.0151,-0.0054,-0.0004,
     *-0.0125,-0.0125,-0.0101,-0.0058,-0.0081,-0.0044, 0.0012,-0.0004,
     * 0.0332, 0.0332, 0.0270, 0.0185, 0.0124, 0.0062, 0.0046,-0.0004,
     * 0.0248, 0.0248, 0.0241, 0.0184, 0.0138, 0.0070, 0.0083,-0.0004,
     *-0.0234,-0.0234,-0.0191,-0.0135,-0.0097,-0.0029,-0.0008,-0.0004/
C
      DATA (((TCN(I,J,K), I=1,8), J=1,13), K=1,1) /
     * 0.    ,-0.0034,-0.0031,-0.0120,-0.0252,-0.0340,-0.0257,-0.0123,
     * 0.    , 0.0008,-0.0026,-0.0143,-0.0242,-0.0344,-0.0232,-0.0123,
     * 0.    , 0.0006, 0.0010,-0.0113,-0.0167,-0.0132,-0.0148,-0.0123,
     * 0.    ,-0.0030,-0.0025,-0.0014,-0.0156,-0.0155,-0.0081,-0.0123,
     * 0.    ,-0.0003,-0.0004,-0.0091,-0.0083,-0.0108,-0.0099,-0.0123,
     * 0.    ,-0.0057,-0.0112,-0.0104,-0.0071,-0.0061,-0.0077,-0.0123,
     * 0.    , 0.0025,-0.0010,-0.0070,-0.0036,-0.0096,-0.0131,-0.0123,
     * 0.    , 0.0023,-0.0072,-0.0072,-0.0101,-0.0088,-0.0090,-0.0123,
     * 0.    , 0.0204,-0.0052,-0.0080,-0.0135,-0.0166,-0.0171,-0.0123,
     * 0.    ,-0.0041,-0.0094,-0.0121,-0.0210,-0.0231,-0.0236,-0.0123,
     * 0.    ,-0.0005,-0.0067,-0.0175,-0.0233,-0.0205,-0.0200,-0.0123,
     * 0.    ,-0.0034,-0.0076,-0.0165,-0.0272,-0.0304,-0.0250,-0.0123,
     * 0.    ,-0.0034,-0.0031,-0.0120,-0.0252,-0.0340,-0.0257,-0.0123/
C
      DATA (((TCN(I,J,K), I=1,8), J=1,13), K=2,2) /
     * 0.    ,-0.0066,-0.0014,-0.0012,-0.0058,-0.0116,-0.0098,-0.0012,
     * 0.    , 0.0018, 0.0015, 0.0005,-0.0047,-0.0116,-0.0070,-0.0012,
     * 0.    ,-0.0035,-0.0052,-0.0035,-0.0057,-0.0060,-0.0063,-0.0012,
```

```
     * 0.    , 0.0024,-0.0014,-0.0005,-0.0044,-0.0049,-0.0008,-0.0012,
     * 0.    , 0.0056, 0.0036,-0.0032,-0.0003, 0.0005, 0.0012,-0.0012,
     * 0.    , 0.0040, 0.0005,-0.0027,-0.0014, 0.0018, 0.0132,-0.0012,
     * 0.    , 0.0051, 0.0013, 0.0002, 0.0017, 0.0028, 0.0001,-0.0012,
     * 0.    , 0.0000,-0.0024,-0.0026,-0.0036,-0.0019, 0.0000,-0.0012,
     * 0.    ,-0.0010, 0.0088,-0.0003, 0.0008,-0.0027,-0.0001,-0.0012,
     * 0.    ,-0.0010, 0.0004,-0.0016,-0.0062,-0.0061,-0.0031,-0.0012,
     * 0.    , 0.0013, 0.0000,-0.0071,-0.0108,-0.0078,-0.0055,-0.0012,
     * 0.    ,-0.0036,-0.0004,-0.0038,-0.0116,-0.0109,-0.0113,-0.0012,
     * 0.    ,-0.0066,-0.0014,-0.0012,-0.0058,-0.0116,-0.0098,-0.0012/
C
      DATA (TBETA(I),I=1,8) / 0.,5.,15.,30.,45.,60.,75.,90. /
C
      DATA (TALPHA(I),I=1,13) / 0.,30.,60.,90.,120.,150.,180.,210.,
     .                          240.,270.,300.,330.,360. /
C
      DATA (TPC(I),I=1,2) / 5.,75. /
C
C ***** CALCULATE THE AERO COEFFICIENTS  *****
C
      CX = TBLU3(BETA,ALPHA,PC,TBETA,TALPHA,TPC,TCX,
     .           1,1,1,8,13,2,8,13,2)
      CY = TBLU3(BETA,ALPHA,PC,TBETA,TALPHA,TPC,TCY,
     .           1,1,1,8,13,2,8,13,2)
      CZ = TBLU3(BETA,ALPHA,PC,TBETA,TALPHA,TPC,TCZ,
     .           1,1,1,8,13,2,8,13,2)
      CL = TBLU3(BETA,ALPHA,PC,TBETA,TALPHA,TPC,TCL,
     .           1,1,1,8,13,2,8,13,2)
      CM = TBLU3(BETA,ALPHA,PC,TBETA,TALPHA,TPC,TCM,
     .           1,1,1,8,13,2,8,13,2)
      CN = TBLU3(BETA,ALPHA,PC,TBETA,TALPHA,TPC,TCN,
     .           1,1,1,8,13,2,8,13,2)
C
      RETURN
      END
```

```
      SUBROUTINE COSDIR(ANG,DCOS)
      DIMENSION ANG(3), DCOS(3,3)
C
C  CALCULATES THE EULER ANGLES FROM THE DIRECTION COSINE MATRIX
C
      ANG(1) = ARTAN2(DCOS(1,2),DCOS(1,1))
C
      ANG(2) = ASIN(-DCOS(1,3))
C
      ANG(3) = ARTAN2(DCOS(2,3),DCOS(3,3))
C
      RETURN
      END
```

AD-A096 597 BOEING MILITARY AIRPLANE CO SEATTLE WA F/G 1/3
ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM. VOLUME --ETC(U)
SEP 80 C L WEST, B R UMMEL, R F YURCZYK F33615-79-C-3407
UNCLASSIFIED AFWAL-TR-80-3014-VOL-1 NL

6 OF 8

AD A
096597

```
      FUNCTION DET3(D11,D12,D13,D21,D22,D23,D31,D32,D33)
C
C      FUNCTION FOR COMPUTING THE VALUE OF A 3 X 3 DETERMINENT
C
      DET3= D11*(D22*D33-D23*D32)-D12*(D21*D33-D23*D31)
     1 + D13*(D21*D32-D22*D31)
      RETURN
      END
```

```
      SUBROUTINE DIRCOS (DCOS,ANG)
      DIMENSION DCOS(3,3), ANG(3)
C
C  DESIGNED BY C.L. WEST
C  LAST MODIFIED -  DECEMBER 6, 1980
C
C  CALCULATES THE DIRECTION COSINE MATRIX FROM THE EULER ANGLES
C
      SINPSI = SIN(ANG(1))
      COSPSI = COS(ANG(1))
      SINTHE = SIN(ANG(2))
      COSTHE = COS(ANG(2))
      SINPHI = SIN(ANG(3))
      COSPHI = COS(ANG(3))
C
      DCOS(1,1) = COSTHE * COSPSI
      DCOS(1,2) = COSTHE * SINPSI
      DCOS(1,3) = -SINTHE
      DCOS(2,1) = SINPHI * SINTHE * COSPSI -
     .               COSPHI * SINPSI
      DCOS(2,2) = SINPHI * SINTHE * SINPSI +
     .               COSPHI * COSPSI
      DCOS(2,3) = SINPHI * COSTHE
      DCOS(3,1) = COSPHI * SINTHE * COSPSI +
     .               SINPHI * SINPSI
      DCOS(3,2) = COSPHI * SINTHE * SINPSI -
     .               SINPHI * COSPSI
      DCOS(3,3) = COSPHI * COSTHE
C
      RETURN
      END
```

```
C      SUBROUTINE DISECT ---- ENTRY POINT OF COMPASS PROG PICKER
          IDENT      PICKER
          LIST       L,R,G,D
          ENTRY      DISECT
          USE        /WORD/
 ITROOP   BSS        4
 ISQUAD   BSS        1
          USE        0
 DISECT   BSS        1
          SB1        -15
          SB2        3
          SA2        ISQUAD
          MX3        -15
 LOOP     BX7        -X2*X3
          BX6        -X7
          SA6        ITROOP+B2
          LX2        X2,B1
          SB2        B2-1
          GE         B2,LOOP
          EQ         DISECT
          END
```

```
      SUBROUTINE EARATE (EADOT,WBODY,EULER)
      DIMENSION EADOT(3),WBODY(3),EULER(3)
      DATA PSID / 0 /
C
C  CALCULATES THE EULER ANGLE RATES FROM THE BODY AXIS ANGULAR
C  VELOCITY VECTOR
C
C  ***************  CALLING SEQUENCE  ***************
C
C  ** OUTPUT **
C
C  EADOT(3)  - EULER ANGLE RATES  -- YAW,PITCH,ROLL --  (RAD/SEC)
C
C  ** INPUT **
C
C  WBODY(3) - X,Y,Z BODY AXIS ANGULAR VELOCITY COMPONENTS (RAD/SEC)
C  EULER(3) - EULER ANGES (RAD)
C
      CP = COS(EULER(2))
      SP = SIN(EULER(2))
      CR = COS(EULER(3))
      SR = SIN(EULER(3))
C
      EADOT(2) = WBODY(2)*CR - WBODY(3)*SR
      IF(CP.NE.0.) PSID = (WBODY(2)*SR + WBODY(3)*CR)/CP
      EADOT(1) = PSID
      EADOT(3) = WBODY(1) + PSID*SP
C
      RETURN
      END
```

```
      FUNCTION FSW(A,B,C,D)
C
C  THIS FUNCTION IS DESIGNED AS FOLLOWS -
C         FSW = B IF A IS LESS THAN ZERO
C         FSW = C IF A IS EQUAL TO ZERO
C         FSW = C IF A IS GREATER THAN ZERO
C
      IF(A) 10,20,30
 10   FSW=B
      GO TO 40
C
 20   FSW=C
      GO TO 40
C
 30   FSW=D
C
 40   RETURN
      END
```

```
      SUBROUTINE LAG (CSDOT,CSCOM,CSPOS,CSTRM,TC,TIME,TO)
C
C  RESPONSE OF A FIRST ORDER LAG FUNCTION TO A CONTROL SURFACE STEP
C  INPUT. TO MAY BE USED TO MECHANIZE A TIME DELAY, WITH THE CONTROL
C  SURFACE REMAINING AT ITS TRIM POSITION UNTIL TIME TO.
C
C  DEFINITION OF CALLING ARGUMENTS ......
C
C      CSDOT - CONTROL SURFACE RATE (DEG/SEC)  ---  OUTPUT  ---
C      CSCOM - CONTROL SURFACE COMMANDED POSITON (DEG)
C      CSPOS - DEFLECTION OF THE CONTROL SURFACE FROM ITS
C              TRIM POSITION (DEG)
C      CSTRM - CONTROL SURFACE TRIM POSITION (DEG)
C      TC    - TIME CONSTANT (SEC)
C      TIME  - SIMULATION TIME (SEC)
C      TO    - TIME DELAY AFTER WHICH THE CONTROL SURFACE RATE IS
C              CALCULATED (SEC)
C
      IF(TIME-TO.GE.0) GO TO 10
      CSDOT = 0
      GO TO 20
C
 10   CSDOT = (CSCOM - (CSPOS+CSTRM))/TC
C
 20   RETURN
      END
```

```
      SUBROUTINE PCAERO (FLIFT,FDRAG,FMDOT,SCT,
     .                   SW,XPC,UPC,TLS,DTI,TDU,VOL,UVL,
     .                   CT,CN,CM,FD,B,STI,RFS,FLA,TLA,TEM)
C
C  THIS ROUTINE DETERMINES PARACHUTE AERODYNAMIC FORCES ACTING ON
C  THE PARACHUTE
C
C  **********  PCAERO OUTPUTS  **********
C
C     FLIFT(3) - X,Y,Z EARTH SYSTEM LIFT COMPONENTS ACTING ON
C                THE PARACHUTE (LB)
C     FDRAG(3) - X,Y,Z EARTH SYSTEM DRAG COMPONENTS ACTING ON
C                THE PARACHUTE (LB)
C     FMDOT(3) - X,Y,Z EARTH SYSTEM MASS ACQUISITION FORCE
C                COMPONENTS ACTING ON THE PARACHUTE (LB)
C     SCT      - COMPUTED TANGENTIAL DRAG AREA (FT**2)
C
C  **********  PCAERO INPUTS  **********
C
C     SW     - FLAG TO INDICATE AERODYNAMIC CALCULATION MODE
C                1 = FROM PARACHUTE LAUNCH TO LINESTRETCH
C                2 = DURING INFLATION
C                3 = DURING REEFING
C                4 = AFTER REEFING
C                5 = PARACHUTE INFLATED
C     XPC(3) - X,Y,Z EARTH SYSTEM LINEAR POSITION VECTOR OF THE
C              PARACHUTE (FT)
C     UPC(3) - X,Y,Z EARTH SYSTEM LINEAR VELOCITY VECTOR OF THE
C              PARACHUTE (FT/SEC)
C     TLS    - TIME AT LINESTRETCH (SEC)
C     DTI    - THE TIME DURATION OF PARACHUTE CANOPY INFLATION (SEC)
C     TDU    - REEFING DURATION (SEC)
C     VOL    - VOLUME OF THE FILLED CANOPY (FT**3)
C     UVL(3) - PARACHUTE LINE UNIT VECTOR
C     CT(3)  - CONSTANTS USED IN THE EQUATION THAT CALCULATES THE
C              TANGENTIAL DRAG AREA
C     CN(3)  - CONSTANTS USED IN THE EQUATION THAT CALCULATES THE
C              NORMAL DRAG AREA
C     CM(2)  - CONSTANTS USED INTHE MACH EFFECTS EQUATION
C     FD     - WAKE TO FREE STREAM RATIO
C     B      - CONSTANT USED IN THE EQUATION FOR CALCULATING
C              SCD OF THE PARACHUTE
C     STI    - INFLATED PARACHUTE DRAG AREA (FT**2)
C     RFS    - PRODUCT OF REFERENCE AREA AND TANGENT FORCE
C              COEFFICIENT WHEN REEFED (FT**2)
C     FLA    - PARACHUTE MODE FLAG
C                  5 = LINES SEVERED
C     TLA    - PARACHUTE LAUNCH TIME / LINE SEVERING TIME (SEC)
C     TEM    - TIME DURATION FOR PARACHUTE EMERGENCE / LINE
C              SEVERENCE (SEC)
C
C  CALLING SEQUENCE DIMENSIONS ......
C
      DIMENSION FLIFT(3),FDRAG(3),FMDOT(3),XPC(3),UPC(3),UVL(3),
     .          CT(3),CN(3),CM(2)
C
C  INTERNAL DIMENSIONS ......
C
```

```
      DIMENSION U0(3),UW(3),TEMP1(3),TEMP2(3),UVV(3),UVLIFT(3)
C
      COMMON /CTIME/ TIME
      COMMON /CIO/ IREAD,IWRITE,IDIAG
C
      DATA PIO2 / 1.57080 /
C
C  ZERO THE MASS ACQUISITION FORCE ......
C
      DO 5 I=1,3
 5    FMDOT(I) = 0
C
C  ----  DETERMINE THE AERO PARAMETERS  ----
C
      CALL ATMOS (VS,RHO,-XPC(3),UW,0,0,0)
      U0(1) = UPC(1) - UW(1)
      U0(2) = UPC(2) - UW(2)
      U0(3) = UPC(3) - UW(3)
      VBAR = SQRT(U0(1)**2+U0(2)**2+U0(3)**2)
      VMACH = VBAR/VS
C
C  ----  CALCULATE ALPHA (THE ANGLE WHOSE COSINE IS THE CHUTE  ----
C        VELOCITY UNIT VECTOR DOTTED ONTO THE LINE UNIT VECTOR
C
C  DETERMINE THE PARACHUTE VELOCITY UNIT VECTOR ......
C
      DO 10 I=1,3
 10   UVV(I) = U0(I)/VBAR
C
C  IF THE LINES HAVE BEEN SEVERED ......
C
      FACTOR = 1.
      IF(FLA.NE.5.) GO TO 15
      IF(TLA.EQ.0) TLA = TIME
      FLIFT(1) = FLIFT(2) = FLIFT(3) = ALPHA = SINA = 0
      COSA = 1.
      DELTA = TIME - TLA
      IF(DELTA.GE.TEM) GO TO 80
      FACTOR = SIN(DELTA*PIO2/TEM)
      GO TO 60
C
C  DOT THE VELOCITY UNIT VECTOR ONTO THE LINE UNIT VECTOR ......
C
 15   CALL DOTPRD (CALPHA,UVV,UVL,3)
      IF(ABS(CALPHA).GE.1.0) CALPHA = SIGN(1.0,CALPHA)
C
      ALPHA = ACOS(CALPHA)
      SINA = SIN(ALPHA)
      COSA = COS(ALPHA)
C
C  ----  CALCULATE THE MASS ACQUISITION FORCE  ----
C
C  CALCULATE THE MASS ACQUISITION FORCE IF SW = 2 OR 4  ......
C
      IF(SW.NE.2. .AND. SW.NE.4.) GO TO 40
C
      RHOS = RHO*((1.+.2*VMACH**2)**2.5)
      PCNTF = ((TIME-TDU)-TLS)/DTI
```

```
      PCNT = PCNTF
      IF (PCNT.GT.0.5) PCNT = 0.5
      DDTM = 0.01*PCNT*VOL*RHOS/DT1
      DO 30 I=1,3
 30   FMDOT(I) = -DDTM*U0(I)
C
C  ***********************************************
C  **     LOGIC TO CHOOSE THE PROPER EQUATIONS    **
C  ***********************************************
C
 40   GO TO (50,60,70,80,60), SW
C
C  ----  EQUATIONS USED PRIOR TO LINESTRETCH  ----
C           AND AFTER THE LINES ARE SEVERED
C
 50   SCT = B * STI
      SCN = 0.0
      GO TO 90
C
C  ----  EQUATIONS USED WHEN THE CHUTE IS INFLATING  ----
C
C  CALCULATE THE WAKE TO FREE STREAM RATIO ......
C
 60   FC = FD
      IF(VMACH.GT.1.0) FC = (1.0+(CM(1)+CM(2)*(VMACH-1.0))*
                            (VMACH-1.0))*FD
C
C  CALCULATE THE VARIABLES USED IN DETERMINING THE NORMAL AND TANGENTIAL
C  DRAG AREAS DURING CHUTE INFLATION ......
C
      SCTIA = STI + ((CT(3)*ALPHA+CT(2))*ALPHA+CT(1))*ALPHA
      SCDLS = B * SCTIA
C
      SCT = SCDLS + (SCTIA-SCDLS)*PCNTF*FC
      SCN = ((CN(3)*ALPHA+CN(2))*ALPHA+CN(1))*ALPHA*PCNTF*FC
      GO TO 90
C
C  ----  EQUATIONS USED WHEN THE CHUTE IS REEFED  ----
C
 70   SCT = RFS
      SCN = 0.
      GO TO 90
C
C  ----  EQUATIONS USED WHEN THE CANOPY IS FILLED  ----
C
 80   SCT = STI + ((CT(3)*ALPHA+CT(2))*ALPHA+CT(1))*ALPHA
      SCN =        ((CN(3)*ALPHA+CN(2))*ALPHA+CN(1))*ALPHA
C
C  ****************************************
C  **  CALCULATE THE LIFT AND DRAG AREAS  **
C  ****************************************
C
 90   SCL = ABS(SCN*COSA - SCT*SINA)
      SCD = ABS (SCN*SINA + SCT*COSA)
C
C  ***********************************************
C  **  CALCULATE THE EARTH AXIS LIFT COMPONENTS  **
C  ***********************************************
```

```
C
C
C  COMPUTE THE UNIT VECTOR IN THE DIRECTION OF LIFT ......
C
      IF(FLA.EQ.5.) GO TO 120
      CALL CRSPRD (TEMP1,UO,UVL)
      CALL CRSPRD (TEMP2,UO,TEMP1)
      RESULT = SQRT(TEMP2(1)**2 + TEMP2(2)**2 + TEMP2(3)**2)
      DO 100 I=1,3
 100  UVLIFT(I) = TEMP2(I)/RESULT
C
      E = .5*RHO*SCL*VBAR*VBAR
C
      DO 110 I=1,3
 110  FLIFT(I) = -E * UVLIFT(I)
C
C  ***********************************************
C  **  CALCULATE THE EARTH AXIS DRAG COMPONENTS  **
C  ***********************************************
C
 120  E = .5*RHO*SCD*VBAR*VBAR
C
      DO 130 I=1,3
 130  FDRAG(I) = -E * UVV(I) * FACTOR
C
      RETURN
      END
```

```
      SUBROUTINE RATIO (NV,GV,TV,RAT,NC)
      DIMENSION TV(1)
C
      IF (NV.EQ.1) GO TO 10
      IF(GV-TV(1).GT.0) GO TO 30
 10   NC = 1
 20   RAT = 0
      GO TO 60
 30   DO 40 NCNT=2,NV
      NC = NCNT
      IF(GV-TV(NC)) 50,20,40
 40   CONTINUE
      GO TO 20
 50   RAT = (TV(NC) - GV)/(TV(NC) - TV(NC-1))
 60   RETURN
      END
```

```
      FUNCTION RLIM(AA,BB,CC)
C
C     FUNCTION WHICH LIMITS THE VALUE OF VARIABLE AA
C     TO WITHIN A RANGE DEFINED BY VARIABLES BB AND CC
C
      IF(AA.LT.BB)GO TO 10
      IF(AA.GT.CC)GO TO 20
      RLIM=AA
      GO TO 30
C
   10 RLIM=BB
      GO TO 30
C
   20 RLIM=CC
C
 30   RETURN
      END
```

```
      SUBROUTINE ROTATEI (BMI,BPI,DC)
      DIMENSION BMI(3),BMIT(3),BPI(3),BPIT(3),DC(3,3)
C
C  TRANSFORMS INERTIA PROPERTIES FROM ONE AXIS SYSTEM
C  TO ANOTHER THROUGH A DIRECTION COSINE MATRIX ......
C
C
C  TRANSFORM THE MOMENTS OF INERTIAS ......
C
      DO 10 I=1,3
      BMIT(I) = DC(I,1)**2*BMI(1) + DC(I,2)**2*BMI(2) +
     .          DC(I,3)**2*BMI(3) - (DC(I,1)*DC(I,2)*BPI(1) +
     .          DC(I,1)*DC(I,3)*BPI(2) + DC(I,2)*DC(I,3)*BPI(3))*2.0
 10   CONTINUE
C
C  TRANSFORM THE PRODUCTS OF INERTIA ......
C
      BPIT(1) = -DC(1,1)*DC(2,1)*BMI(1) - DC(1,2)*DC(2,2)*BMI(2) -
     .           DC(1,3)*DC(2,3)*BMI(3) + (DC(1,1)*DC(2,2) +
     .           DC(1,2)*DC(2,1))*BPI(1) + (DC(1,1)*DC(2,3) +
     .           DC(1,3)*DC(2,1))*BPI(2) + (DC(1,2)*DC(2,3) +
     .           DC(1,3)*DC(2,2))*BPI(3)
      BPIT(2) = -DC(1,1)*DC(3,1)*BMI(1) - DC(1,2)*DC(3,2)*BMI(2) -
     .           DC(1,3)*DC(3,3)*BMI(3) + (DC(1,1)*DC(3,2) +
     .           DC(1,2)*DC(3,1))*BPI(1) + (DC(1,1)*DC(3,3) +
     .           DC(1,3)*DC(3,1))*BPI(2) + (DC(1,2)*DC(3,3) +
     .           DC(1,3)*DC(3,2))*BPI(3)
      BPIT(3) = -DC(2,1)*DC(3,1)*BMI(1) - DC(2,2)*DC(3,2)*BMI(2) -
     .           DC(2,3)*DC(3,3)*BMI(3) + (DC(2,1)*DC(3,2) +
     .           DC(2,2)*DC(3,1))*BPI(1) + (DC(2,1)*DC(3,3) +
     .           DC(2,3)*DC(3,1))*BPI(2) + (DC(2,2)*DC(3,3) +
     .           DC(2,3)*DC(3,2))*BPI(3)
      DO 20 I=1,3
      BMI(I) = BMIT(I)
 20   BPI(I) = BPIT(I)
C
      RETURN
      END
```

```
      FUNCTION TBLU3(X1,Y1,Z1,X,Y,Z,F3,NDX,NDY,NDZ,NX,NY,NZ,MX,MY,MZ)
C******
C     PURPOSE
C        TBLU3 PERFORMS TABLE SEARCH AND LAGRANGIAN POLYNOMIAL
C        INTERPOLATION OF USER-DEFINED DEGREE ON 3 INDEPENDENT
C        VARIABLES
C     USAGE
C        DIMENSION X(NX),Y(NY),Z(NZ),F3(MX,MY,MZ)
C        V = TBLU3(X1,Y1,Z1,X,Y,Z,F3,NDX,NDY,NDZ,NX,NY,NZ,MX,MY,MZ)
C     INPUT PARAMETERS
C          X1,Y1,Z1 - POINT TO INTERPOLATE FOR
C             X,Y,Z - ARRAYS OF INDEPENDENT VARIABLES
C                F3 - 3D ARRAY OF DEPENDENT VARIABLE
C       NDX,NDY,NDZ - DEGREE OF INTERPOLATION FOR EACH DIMENSION
C          NX,NY,NZ - IABS OF EACH IS THE NUMBER OF DATA POINTS IN
C                     THE RESPECTIVE X, Y OR Z ARRAY.  IF NEGATIVE,
C                     NEAREST END POINT IS TO BE USED UPON
C                     EXTRAPOLATION
C          MX,MY,MZ - DIMENSIONAL CONSTANTS FOR F3 ARRAY
C     OUTPUT PARAMETERS
C        V - RESULT OF TABLE SEARCH AND INTERPOLATION
C            SUCCESS  V = INTERPOLATED VALUE
C              ERROR  V = INDEFINITE VALUE WHERE RIGHTMOST DIGIT
C                         DEFINES THE ERROR DETECTED
C                1   DATA VALUES WITHIN X, Y OR Z ARE NOT DISTINCT
C                2   ONE OF NDX, NDY OR NDZ IS LESS THAN ZERO
C                3   ONE OF NX, NY OR NZ IS ZERO
C                4   EITHER MX.LT.IABS(NX) OR MY.LT.IABS(NY)
C******
C
      DIMENSION X(1),Y(1),Z(1),F3(MX,MY,MZ)
      INTEGER SEARCH
      DATA ERR2/17770000000000000002B/
      DATA ERR3/17770000000000000003B/
      DATA ERR4/17770000000000000004B/
C        TEST FOR USER ERRORS
      TBLU3 = 0
      IF ((NDX.LT.0).OR.(NDY.LT.0).OR.(NDZ.LT.0)) TBLU3 = ERR2
      IF ((NX.EQ.0).OR.(NY.EQ.0).OR.(NZ.EQ.0)) TBLU3 = ERR3
      IF ((MX.LT.IABS(NX)).OR.(MY.LT.IABS(NY))) TBLU3 =ERR4
      IF (TBLU3.NE.0) GO TO 50
C        SET UP INITIAL PARAMETERS
      X2 = X1
      Y2 = Y1
      Z2 = Z1
      MDX = NDX
      MDY = NDY
      MDZ = NDZ
C        SEARCH FOR X1, Y1 AND Z1 IN TABLES
      IX = SEARCH(X2,X,MDX,NX,I)
      IY = SEARCH(Y2,Y,MDY,NY,J)
      IZ = SEARCH(Z2,Z,MDZ,NZ,K)
C        TEST FOR EXACTNESS IN 1 OR MORE DIMENSIONS
      IW = IX+IY+IZ
      IF (IW.EQ.0) GO TO 40
      IF (IW.NE.3) GO TO 10
      TBLU3 = F3(I,J,K)
      GO TO 50
```

```
   10 IF (IX.EQ.0) GO TO 20
      X2 = X(I)
      MDX = 0
   20 IF (IY.EQ.0) GO TO 30
      Y2 = Y(J)
      MDY = 0
   30 IF (IZ.EQ.0) GO TO 40
      Z2 = Z(K)
      MDZ = 0
C         INTERPOLATE
   40 TBLU3 = TERP3(X2,Y2,Z2,X,Y,Z,F3,MDX,MDY,MDZ,MX,MY,MZ,I,J,K)
   50 RETURN
      END
```

```
      SUBROUTINE TLU(IB,NX,NY,NZ,ROW,COLM,PAGE,XG,YG,ZG,ANS,NTAB)
      DIMENSION IB(1),ROW(NX),COLM(NY),PAGE(NZ),ANS(6)
      FIRSTF (X,Y,Z) = X - Z*(X - Y)
C
C  WHAT BALL PARK IS THE POINT IN  .......
C
      CALL RATIO (NX,XG,ROW,RATX,I)
      CALL RATIO (NY,YG,COLM,RATY,J)
      CALL RATIO (NZ,ZG,PAGE,RATZ,K)
C
C  IT'S JUST PRIOR TO THE 'ITH' ROW, 'JTH' COLUMN AND THE 'KTH' PAGE ...
C
C
C  NOTE -  IF ONE OF THE INCOMING ARGUMENTS IS OUT OF THE TABLE BOUNDS,
C          THE APPROPRIATE VALUE OF RATX,RATY,RATZ, WILL BE ZERO .....
C
C  WHAT IS THE LOCATION OF THE NEXT HIGHER POINT  .....
C
      NXY = NX*NY
      NP = I + NX*(J-1) + NXY*(K-1)
C
C  LET'S INTERPOLATE FROM AS MANY AS NTAB TABLES.....
C
      DO 50 L=1,NTAB
C
      B = C = E = 0
C
C  WHERE IS THE POINT BETWEEN ROWS  ....
C
      CALL UNPACK (NP,IB,C1,C2)
      A = FIRSTF (C2,C1,RATX)
C
C  IF WE ARE IN THE FIRST COLUMN JUMP TO STATEMENT 10 .....
C
      IF (J.EQ.1) GO TO 10
C
C  JUMP TO THE NEXT LOWER COLUMN .....
C
      NP = NP - NX
C
C  BETWEEN ROWS IN THE ADJACENT COLUMN .....
C
      CALL UNPACK (NP,IB,C1,C2)
      B = FIRSTF (C2,C1,RATX)
C
C  JUMP TO THE NEXT LOWER PAGE.....
C
 10   NP = NP - NXY
C
C  IF WE ARE NOT ON THE FIRST PAGE JUMP TO STATEMENT 20 .....
C
      IF (K.NE.1) GO TO 20
C
C  IF WE ARE IN THE FIRST COLUMN, JUMP TO STATEMENT 40 .....
C
      IF (J.EQ.1) GO TO 40
C
C  JUMP TO THE NEXT HIGHER COLUMN .....
```

```
C
      NP = NP + NX
      GO TO 40
C
C  IF WE ARE IN THE FIRST COLUMN, JUMP TO STATEMENT 30 .....
C
 20   IF (J.EQ.1) GO TO 30
C
C  BETWEEN ROWS .....
C
      CALL UNPACK (NP,IB,C1,C2)
      C = FIRSTF (C2,C1,RATX)
C
C  JUMP TO THE NEXT HIGHER COLUMN .....
C
      NP = NP + NX
C
C  BETWEEN ROWS AGAIN IN THE NEXT HIGHER COLUMN ....
C
 30   CALL UNPACK (NP,IB,C1,C2)
      D = FIRSTF (C2,C1,RATX)
C
C  BETWEEN COLUMNS .......
C
      E = FIRSTF (D,C,RATY)
 40   F = FIRSTF (A,B,RATY)
C
C  NOW BETWEEN PAGES .....
C
      ANS(L) = FIRSTF (F,E,RATZ)
C
C  MOVE TO THE BEGINNING OF THE NEXT TABLE ....
C
 50   NP = NP + NXY + NZ*NXY
C
C  THAT'S IT, LET'S GO HOME ....
C
      RETURN
      END
```

```
      SUBROUTINE UNPACK (NP,IB,WORD1,WORD2)
      DIMENSION  IB(1),JWORD(8)
      COMMON /WORD/ ITROOP(4),ISQUAD
      DATA FOUR /4.0/, BN /16383./,CMIN /-1.5/, RANGE /3.0/
      NPRIOR = 1
      PNTS = NP
      WORD = PNTS/FOUR
      NWORD = WORD + 0.1
      NSUBWRD = (WORD - FLOAT (NWORD))*FOUR
      IF (NSUBWRD.EQ.1) NPRIOR = 2
      IF(NPRIOR.EQ.1) GO TO 20
      ISQUAD = IB(NWORD)
      CALL DISECT
      DO 10 I=1,4
10    JWORD(I) = ITROOP(I)
20    ISQUAD = IB(NWORD+1)
      IF (NSUBWRD.EQ.0) ISQUAD = IB (NWORD)
      CALL DISECT
      DO 30 I=1,4
30    JWORD(I+4) = ITROOP(I)
      IF (NSUBWRD.EQ.0) NSUBWRD = 4
      IWORD1 = JWORD(NSUBWRD+3)
      IWORD2 = JWORD(NSUBWRD+4)
      WORD1 = CMIN + (FLOAT(IWORD1)/BN)*RANGE
      WORD2 = CMIN + (FLOAT(IWORD2)/BN)*RANGE
      RETURN
      END
```

```
      SUBROUTINE VECXYZ (TRANS,VEC,ORIGIN,DC,IOPT)
      DIMENSION TRANS(3),VEC(3),ORIGIN(3),DC(3,3),DIFF(3)
C
C  TRANSFORMS VECTORS FROM ONE REFERENCE FRAME INTO ANOTHER ......
C
C           *****  CALLING ARGUMENTS *****
C
C     TRANS(3)    - TRANSFORMED VECTOR (OUTPUT)
C     VEC(3)      - INPUT VECTOR
C     ORIGIN(3)   - SECONDARY SYSTEM ORIGIN IN THE PRIMARY SYSTEM
C     DC(3,3)     - DIRECTION COSINE MATRIX
C     IOPT        - FLAG TO DETERMINE TYPE OF TRANSFORMATION
C                          1 = FROM PRIMARY TO SECONDARY
C                          2 = FROM SECONDARY TO PRIMARY
C
      IF(IOPT.EQ.2) GO TO 20
C
      DO 10 I=1,3
 10   DIFF(I) = VEC(I) - ORIGIN(I)
      CALL MATMPY (TRANS,DC,DIFF,3,3,1)
      GO TO 40
C
 20   CALL MATMPY (TRANS,DC,VEC,3,3,1)
      DO 30 I=1,3
 30   TRANS(I) = ORIGIN(I) + TRANS(I)
C
 40   RETURN
      END
```

```
      SUBROUTINE VELXYZ (U,USEC,XPT,WSEC,DSI)
      DIMENSION U(3),USEC(3),XPT(3),WSEC(3),UPTSEC(3),
     .          TEMP(3),DSI(3,3)
C
C  COMPUTES THE EARTH SYSTEM VELOCITY VECTOR OF A POINT
C  DISPLACED FROM THE ORIGIN OF A SECONDARY COORDINATE SYSTEM
C
C  **********  CALLING ARGUMENTS  **********
C
C     U(3)     - X,Y,Z EARTH SYSTEM VELOCITY VECTOR OF A POINT
C                DISPLACED FROM THE ORIGIN OF A SECONDARY SYSTEM
C                (FT/SEC)    --  OUTPUT  --
C     USEC(3)  - X,Y,Z BODY AXIS VELOCITY VECTOR OF THE SECONDARY
C                SYSTEM (FT/SEC)
C     XPT      - X,Y,Z BODY AXIS POSITION VECTOR OF THE DISPLACED
C                POINT IN THE SECONDARY SYSTEM (FT)
C     WSEC     - X,Y,Z BODY AXIS ANGULAR VELOCITY VECTOR OF THE
C                SECONDARY SYSTEM (RAD/SEC)
C     DSI(3,3) - SECONDARY TO EARTH SYSTEM DIRECTION COSINE MATRIX
C
C
C  CALCULATE WSEC X XPT ......
C
      CALL CRSPRD (TEMP,WSEC,XPT)
C
C  DETERMINE USEC + (WSEC X XPT) ......
C
      DO 10 I=1,3
   10 UPTSEC(I) = USEC(I) + TEMP(I)
C
C  TRANSFORM THE VELOCITY VECTOR FROM THE SECONDARY TO THE
C  EARTH SYSTEM ......
C
      CALL MATMPY (U,DSI,UPTSEC,3,3,1)
C
      RETURN
      END
```

APPENDIX I

FILOAD INPUT DATA

This Appendix contains the FILOAD Input Data. FILOAD is a program which creates a random access file from input data that defines the variable names in the calling sequence for each standard component. This random access file is employed by the Model Generation program to build the model defined by the user in the Model Generation input data.

```
NEW FILE
FILE NAME=EASIEST
ABINPT   =    8
WT         BMI  3    BPI  3    FAB  3    TAB  3    FAU  3    TAU  3    TRM  3
ABOUTP   =    4
UAB  3    SXAB  3   SWAB  3   SEAB  3   S
ABTABS   =    0
SYMBOL, AB =   101
AEINPT   =   29
AW         B         C         S         XCP       ZCP       AMI  3    API  3
THR        AIL       ELE       RUD       XEN  3    END  3    TAL       TVE
FRA  3   1 TRA  3  1 FCA  3  1 TCA  3  1 FDA  3  1 TDA  3  1 FRA  3  2 TRA  3
FCA  3   2 TCA  3  2 FDA  3  2 TDA  3  2 CPF
AEOUTP   =    9
UAP  3    SXAP  3   SWAP  3   SEAP  3   STRM  4   SALP      BET       VM
ALT
AETABS   =    0
SYMBOL, AE =   101
AFINPT   =    6
C00        C1        C2        C3        C4        C5
AFOUTP   =    1
S          2
AFTABS   =    0
SYMBOL, AF =   101
AGINPT   =    5
H          WIN  3    BP        TE        SW
AGOUTP   =    2
VS         RHO
AGTABS   =    0
SYMBOL, AG =   101
AMINPT   =   14
FL         PRT       EXP       GXP       GXN       GYL       GZL       DRP
DRN        RDL       DR        GX        GY        GZ
AMOUTP   =    4
DRE        RAD       PTS       PTI
AMTABS   =    0
SYMBOL, AM =   101
APINPT   =   11
UP         XPC  3    PA        EPL  3    ZEM       SRP  3    UST  3    EST  3
WST  3     XAP  3    EAP  3
APOUTP   =    6
F2   3   1 T2   3  1 SW        ALP       CX        CZ
APTABS   =    2
TCX        20.   1
TCZ        20.   1
SYMBOL, AP =   101
ASINPT   =   19
OFF        UP        ZWS       XEM  3    COX       ECX       ECY       ECZ
CLP        CMQ       CNR       S         SRP  3    UST  3    EST  3    WST  3
OSA  3 3   SRA  3    RON
ASOUTP   =   17
F2   3   1 T2   3  1 ALP       BET       VM        Q         CX        CY
CZ         CL        CM        CN        EXL       EXA       CEN  3    TCZ 20
HO
ASTABS   =    1
TAE        20.   1
SYMBOL, AS =   101
AVINPT   =   14
```

```
U     3      W     3      ALT          EA    3      ID          1 VS           ALS          S
UW           VW           WW           PW           QW            RW
AVOUTP   =    18
UG    3      WG    3      ID         2 QW         2 RW          2 CAL          SAL          AL
ALP          VT           BE           WP           UP            EU    3      SIG          QC
QS           MAC
AVTABS   =     0
SYMBOL, AV =    101
CEINPT   =    16
SW           PC           CEW          CMI   3      CPI   3       CLP          CMQ          CNR
XSP   3      FAB   3      TAB   3      FDO   3      TDO   3       FAU   3      TAU   3      TRM   3
CEOUTP   =    29
UCP   3      SXCP  3      SWCP  3      SECP  3      SSCD          SSC          SGX          GY
GZ           DR           FAD   3      TAD   3      WT            S            B            C
CIN   4      CX           CY           CZ           CL            CM           CN           ALP
BET          VM           Q            ALT          SEP
CETABS   =     0
SYMBOL, CE =    101
CGINPT   =    20
W     3      IR           TMI          AD           DA            GR           BDZ          CS
CA           PLD          DG           GFI          DR            DRS          CB           CBS
DBZ          DF           DS           H
CGOUTP   =     8
AL           SALD         AX           SSG          SSGD          SGI          SSF          ST    3
CGTABS   =     0
SYMBOL, CG =    101
CSINPT   =    10
COA          TCA          TDA          COE          TCE           TDE          COR          TCR
TOR          TRM   4
CSOUTP   =     3
AIL          SELE         SRUD         S
CSTABS   =     0
SYMBOL, CS =    101
CTINPT   =    31
SW           UP           SAP   3      AAP   3      UCL           CSK          VI           PA
PT           CBP          C            CI           PMW           SK           CK           GAM
TF           C1           C2           B            BXP           TI           TDE          SRP   3
UST   3      EST   3      WST   3      XAP   3      UAP   3       EAP   3      WAP   3
CTOUTP   =    19
EF           SEL          SWK          SWB          SFL           FON          FCA   3    1 TCA   3
F1    3    1 T1    3    1 CF           CEX          CV            TLO          PC           R
CVH          TSO          FSO
CTTABS   =     1
TCP          20.     1
SYMBOL, CT =    101
DEINPT   =     3
S     N    1 TAU          N        1 1
DEOUTP   =     2
S     N    ZAD1     N     S
DETABS   =     0
SYMBOL, DE =    -99
MODES = DE
DFINPT   =     9
S     N    1 Z0     N     Z1     N     Z2     N     P0     N     P1     N     TAU          CPU
N     1  1
DFOUTP   =     8
S     N    ZAD1     N     SDZ    N     SA0    N     A1     N     A2     N     B1     N     B2     N
DFTABS   =     0
```

```
SYMBOL, DF =    -99
MODES = DF
DIINPT   =     3
S     N   1 S     N   3 N     1 1
DIOUTP   =     1
S     N   2
DITABS   =     0
SYMBOL, DI =    101
MODES = DI
DRINPT   =     6
DAP   3    DBA   3     XAP   3     EAP   3     SRP   3     EST   3
DROUTP   =     7
F2    3   1 T2     3   1 FDA   3   1 TDA   3   1 DLL         DBF         SW
DRTABS   =     1
TBF        20.     1
SYMBOL, DR =    101
ENINPT   =     6
TCO        THR         GAX         GAZ         XO          ZO
ENOUTP   =     3
TH         SF     3    T     3
ENTABS   =     0
SYMBOL, EN =    101
FMINPT   =     5
T     3    GAI         DMP         WN          CX    3
FMOUTP   =     2
A          SAD         S
FMTABS   =     0
SYMBOL, FM =    101
FPINPT   =     3
A          AD          CW    3
FPOUTP   =     2
EA    3   2 W     3
FPTABS   =     0
SYMBOL, FP =    101
FUINPT   =     2
S         1 AN
FUOUTP   =     1
S         2
FUTABS   =     1
FTA        46.     1
SYMBOL, FU =    101
FVINPT   =     4
S         1 S         3 AN          BN
FVOUTP   =     1
S         2
FVTABS   =     1
FTA       172.     2
SYMBOL, FV =    101
FWINPT   =     6
S         1 S         3 S         4 ANX         ANY         ANZ
FWOUTP   =     1
S         2
FWTABS   =     1
FTA       242.     3
SYMBOL, FW =    101
F2INPT   =    11
EDI        EQI         ADI         AQI         AB          RNL         RS1         RW
XW         XC          WO
```

```
F2OUTP   =    10
EDO       SEQO      SADO      SAGO      SARO      A1O       RTF       PHI
AT        PF
F2TABS   =     0
SYMBOL, F2 =   101
GPINPT   =    18
SW        UV   3    XMO  3    XYZ  3    EA   3    XR        XD        ER   3
ED   3    TDE       SRP  3    UST  3    EST  3    WST  3    XPP  3    UPP  3
EPP  3    WPP  3
GPOUTP   =    13
FL        FMT       F1   3  1 T1   3  1 FPP  3    TPP  3    TIN       TLA
FSO  3    TSO  3    FPO  3    TPO  3    TRM  3
GPTABS   =     1
TMF       20.   1
SYMBOL, GP =   101
G8INPT   =    28
W         E1        E2        E8        PSM       X1        X2        X3
X4        X5        X6        R1        R2        R3        R4        WO
K1        K2        X7        X8        AFB       EFB       T1        R5
AM        AN        K3        K4
G8OUTP   =    18
SMC       SA4       SA5       SSD       SSQ       SE5       SE7       SA1
A2        A3        A7        SM        E3        E4        E6        A9
SN        TD
G8TABS   =     2
FA1       45.   1
FE4       45.   1
SYMBOL, G8 =   101
HGINPT   =     4
S       1 FUP       FLO       STR
HGOUTP   =    17
F1        F2        F3        F4        F5        F6        F7        F8
F9        F10       F11       F12       F13       F14       F15       F16
FA
HGTABS   =     0
SYMBOL, HG =   101
HQINPT   =    19
OFF       UP        ZWS       XEM  3    CDX       ECX       ECY       ECZ
CLP       CMQ       CNR       S         SRP  3    UST  3    EST  3    WST  3
DSA  3 3  SRA  3    RON
HQOUTP   =    17
F2   3  1 T2   3  1 ALP       BET       VM        Q         CX        CY
CZ        CL        CM        CN        EXL       EXA       CEN  3    TCZ 20
HD
HQTABS   =     1
TAE       20.   1
SYMBOL, HQ =   101
HYINPT   =     4
S    N  1 GAI  N    DEL  N    N    1 1
HYOUTP   =     5
S    N  2 SL   N    CU   N    TL        CPU
HYTABS   =     0
SYMBOL, HY =   101
MODES = HY
ININPT   =     3
S    N 11 GKI  N 1  N    1 1
INOUTP   =     1
S    N  2S
```

```
INTABS   =     0
SYMBOL, IN =    101
MODES = IN
ITINPT   =     6
S     N   1 GK1   N      GKL   N      AMA   N      AM1   N      N      1 1
ITOUTP   =     1
S     N   2S
ITTABS   =     0
SYMBOL, IT =    101
MODES = IT
IIINPT   =     4
ER         W1            W1            WO
IIOUTP   =     5
TA         SE1           E2            WA            TX
IITABS   =     0
SYMBOL, II =    101
LAINPT   =     4
S     N   1 GAI   N      TC     N      N      1 1
LAOUTP   =     1
S     N   2S
LATABS   =     0
SYMBOL, LA =    101
MODES = LA
LDINPT   =    37
YB         YBD           YP            YR            YDR           YDA           LB            LBD
LP         LR            LDR           LDA           NB            NBD           NP            NR
NDR        NDA           RUD           AIL           UD            WD            F     3       T     3
MA         B             XP            ID            CAL           SAL           UO    3       WO    3
BE         EU    3       VT            QS            RW
LDOUTP   =     4
FY         2  VO         TX          2 TZ          2
LDTABS   =     0
SYMBOL, LD =    101
LEINPT   =     5
S     N   1 GAI   N      ZO     N      PO     N      N      1 1
LEOUTP   =     2
X1    N    SS     N   2
LETABS   =     0
SYMBOL, LE =    101
MODES = LE
LGINPT   =     4
S     N   1  ZO     N      PO     N      N      1 1
LGOUTP   =     1
S     N   2S
LGTABS   =     0
SYMBOL, LG =    101
MODES = LG
LIINPT   =    23                                                                              FTR
UFF        BLI           APX   3       AP1   3       AP2   3       AP3   3       AP4   3       UDO   3
FSO        ULL           ULS           GUR           TYP           FL            XDO   3
EDO   3    WDO   3       XPP   3       UPP   3       EPP   3       XPC   3       UPC   3
LIOUTP   =    28                                                                              FAP   3
EC         STF           SFLA          SW1           FDO   3       TDO   3       FLP   3       CON   4
VAP   3    FLL           ELM           ELC           DEM           RMN           DIS           CWT
TCG  20    UVL   3       RL            RLO           VL            VCG           PCG
TPE        PVL           TLS           VLS
LITABS   =     1
TCW        20.    1
```

```
SYMBOL, LI =    101
LLINPT    =     5
S     N   1 TC1   N      TC2   N     GAI   N      N      1 1
LLOUTP    =    2
X1    N    SS     N   2
LLTABS    =     0
SYMBOL, LL =    101
MODES = LL
LOINPT    =    35
XO          XA          XU          XDE         ZO          ZA          ZAD         ZQ
ZU          ZDE         MO          MAL         MAD         MQ          MU          MDE
MA        1 C           XP        1 ID          CAL         SAL         F         3 T         3
ELE         AL          ALP         UO        3 UP          WP          VT          QS
WO      3   QW          EU      3
LOOUTP    =     7
FX        2 FZ        2 TY        2 UD          WD          MA        2 XP        2
LOTABS    =     0
SYMBOL, LO =    101
L2INPT    =     8
ADI         AQI         RS2         RL          RNL         XL          XC          WO
L2OUTP    =     5
EDO       SEQO        SADS        SAQS        SRTL
L2TABS    =     0
SYMBOL, L2 =    101
MAINPT    =     4
S     N   1 C1    N      C2    N     N      1 1
MAOUTP    =     1
S     N   2
MATABS    =     0
SYMBOL, MA =    101
MODES = MA
MCINPT    =     8
S     N   1 S     N   3 S     N   4 C1    N     C2    N     C3    N     C4    N     N      1
MCOUTP    =     1
S     N   2
MCTABS    =     0
SYMBOL, MC =    101
MODES = MC
MDINPT    =     5
S     N     DMP   N      WN    N     GAI   N      N      1 1
MDOUTP    =     3
Q     N    SQD    N    SQDD    N
MDTABS    =     0
SYMBOL, MD =    101
MODES = MD
MEINPT    =     3
F     3     DCM   N 3   N      1 1
MEOUTP    =     1
S     N   2
METABS    =     0
SYMBOL, ME =    101
MODES = ME
MFINPT    =     5
S     N   1 S     N   3 S     N   4 S     N   5 N      1 1
MFOUTP    =     1
S     N   2
MFTABS    =     0
SYMBOL, MF =    101
```

```
MODES = MF
MGINPT    =      2
PLG   4      1M
MGOUTP    =      1
AMG   4
MGTABS    =      0
SYMBOL, MG =    101
MMINPT    =      5
Q     N      QD     N      QDD   N      PCM   3 N   N      1 1
MMOUTP    =      3
X     3      XD     3      XDD   3
MMTABS    =      0
SYMBOL, MM =    101
MODES = MM
MPINPT    =     32
SW           XYZ  3      EA     3      XR            XD            ER    3      ED    3      UV    3
CSK          VI          PA            PT            CBP           C            CI           PMW
GAM          TF          C1            C2            B             BXP          TI           TOE
SRP   3      UST  3      EST    3      WST   3       XPP   3       UPP   3      EPP   3      WPP   3
MPOUTP    =     19
EF           SEL         SWK           SWB           SFL           F1    3   1  T1    3   1  FPP   3
IPP   3      FM          EXM           VM            TLO           PC           R            CVH
TSO          FSO         TRM    3
MPTABS    =      1
TMP          20.     1
SYMBOL, MP =    101
MRINPT    =      4
TC           AL          GTM           TMS
MROUTP    =      1
TR
MRTABS    =      1
COE          29.     1
SYMBOL, MR =    101
MTINPT    =      3
S     N   1 S     N   3 N      1 1
MTOUTP    =      1
S     N   2
MTTABS    =      0
SYMBOL, MT =    101
MODES = MT
M1INPT    =      4
U     3      W      3      EA     3      RFL
M1OUTP    =      4
PDV          ALP           MAC           FLG
M1TABS    =      0
SYMBOL, M1 =    101
M2INPT    =     17
FLG          PDV           CX            LP            NN            LP           L            MQ
M            NR            MMP           RHO           RFL           RFA          MA           IXX
IYY
M2OUTP    =     19
U     3      SEA    3      SW     3      SFX           FY            FZ           TX           TY
TZ           XD            YD            ZD            X             Y            Z            RPM
PIT          YAW           TLT
M2TABS    =      0
SYMBOL, M2 =    101
SCINPT    =      0
SCOUTP    =      0
```

```
OCTABS    =     0
SYMBOL, OC =    400
PCINPT    =    30
STI        RSC         RFM         RFD          RFS          B           CI          CT   3
CN   3     CM    2     FD          PWT          PMI    3     PPI   3     TEM         CSP
CUP        DPG   3     FLA         FLP    3     FPP    3     TPP   3     VAP   3     UVL  3
RL         VCG         PCG         CWT          TPE          TRM   3
PCOUTP    =    19
UPP  3    SXPP   3    SWPP   3    SEPP   3    SUPC   3    SXPC   3    SPHA        SW
FLI  3     FDR   3     FMA   3     RM           VOL          TLA         TLS         TDS
DTI        TDU         TRF
PCTABS    =     0
SYMBOL, PC =    101
PFINPT    =    14
ED         EQ          AD          AQ           X1           X2          X3          X4
PFR        AB          VB          CMA          CMI          G1
PFOUTP    =     9
B1        SB2         SARO         AIO          AT           VPF         PFL         FIN
FO
PFTABS    =     0
SYMBOL, PF =    101
PMINPT    =    10
F    3     MA          LA          LO           ALT          TI          DA          VEL
AZI        GAM
PMOUTP    =     2
R    3    SRD    3    S
PMTABS    =     0
SYMBOL, PM =    101
POINPT    =     5
R    3     RD    3    A      3 3   TI           DA
POOUTP    =     6
LA         LO         ALT          AZI          GAM          EA    3
POTABS    =     0
SYMBOL, PO =    101
RAINPT    =     0
RAOUTP    =     4
NU         NV          NW          NP
RATABS    =     0
SYMBOL, RA =    101
RGINPT    =     4
W    3   1 SL         DMP          WN
RGOUTP    =     2
W    3   2SWX    3    S
RGTABS    =     0
SYMBOL, RG =    101
RKINPT    =     4
FON        XRN   3     YAW         PIT
RKOUTP    =     6
PHA        RON         FST   3     TST    3     FR           TIG
RKTABS    =     1
TRF        20.    1
SYMBOL, RK =    101
RLINPT    =    26
BL1  3     BL2   3     BL3   3     BL4    3     BL5    3     BL6   3     UP          RLR
ARR  3     RLL         XRL   3     ERL    3     SPR    2     DPG   2     SBF         ZTS
BTS        CPT   3     SRP   3     UST    3     EST    3     WST   3     XAP   3     UAP  3
EAP  3     WAP   3
RLOUTP    =    12
```

```
F2    3   1  T2     3   1  FRA   3   1  TRA   3   1  FL          FTS         TTS         OFF
DSA   3  3   SRA    3      DIS          TM    3
RLTABS    =     0
SYMBOL, RL =    101
RNINPT    =     3
AX           SIG         MN
RNOUTP    =     1
S         2
RNTABS    =     0
SYMBOL, RN =    101
RSINPT    =    15
FL           XYZ    3    EA     3    XPB   3    UPB   3    EPB   3    WPB   3    XAB   3
UAB   3      EAB    3    WAB    3    XR         XD         ER    3    ED    3
RSOUTP    =     5
FPB   3      TPB    3    FAB    3    TAB   3    TRM   3
RSTABS    =     0
SYMBOL, RS =    101
SAINPT    =     8
S     N   1  C1     N    C2     N    C3    N    C4    N    C5    N    C6    N    N     1
SAOUTP    =     1
S     N   2
SATABS    =     0
SYMBOL, SA =    101
MODES  =  SA
SCINPT    =    17
SW           UP          SAP    3    AAP   3    UCL        CSK         SK          CK
IDE          SRP    3    UST    3    EST   3    WST   3    XAP   3    UAP   3    EAP   3
WAP   3
SCOUTP    =    12
FL           FUN         FCA    3    TCA   3    FCS   3    TCS   3    CF          CEX
CV           TCT         TSO         FSO
SCTABS    =     1
TCF          20.    1
SYMBOL, SC =    101
SDINPT    =    12
UD           VD          WD          TX         TY         TZ          IXX         IYY
IZZ          IXZ         IXY         IYZ
SDOUTP    =     8
G     3    SW       3    SEA    3   SXD         YD         ALR         ALT         SWD   3
SDTABS    =     0
SYMBOL, SD =    101
SEINPT    =    37
F1    3   1  F1     3   2  F1     3   3  F1    3   4  F1    3   5  F1    3   6  F1    3   7  F1    3
T1    3   1  T1     3   2  T1     3   3  T1    3   4  T1    3   5  T1    3   6  T1    3   7  T1    3
F2    3   1  F2     3   2  F2     3   3  F2    3   4  F2    3   5  F2    3   6  F2    3   7  F2    3
T2    3   1  T2     3   2  T2     3   3  T2    3   4  T2    3   5  T2    3   6  T2    3   7  T2    3
CW           CCG    3    CM1    3    CPI   3    TM    3
SEOUTP    =    11
GST   3     SSRP    3   SWST    3   SEST   3   SSCD        SSC         SGX         GY
GZ           DR          ALT
SETABS    =     0
SYMBOL, SE =    101
SGINPT    =     4
FR1          FR2         AM1         AM2
SGOUTP    =     4
S            F           LGF         AMP
SGTABS    =     0
SYMBOL, SG =    101
```

```
SLINPT   =     2
UD   3    WD   3
SLOUTP   =     4
UAP  3    SXAP  3   SWAP  3   SEAP  3   S
SLTABS   =     0
SYMBOL, SL =   101
SPINPT   =    20
FL        YPR        AVW        WMI        SMI        RII        RIF        XR   3
GV   3    GSA        GSF        SPR        DPG        FMT        TMX        TNF
TOS       TSU        GMA        WST  3
SPOUTP   =     8
WG        SESG  3    SESR  3    SPHA       F1    3  1 T1    3  1 TIN        ECA
SPTABS   =     3
TRI       20.    1
TMA       20.    1
TST       20.    1
SYMBOL, SP =   101
SRINPT   =     9
FGN       PCG  3     EA    3    XRN  3     YAW        PIT        PL         POD
PID
SROUTP   =    15
W        1SPHA       RON        F1   3   1 T1    3  1 X     3  1 BM    3  1 BP   3
FR        PWI        SPI        RHO        VWI        TM1  3     TIG
SRTABS   =     1
TRF       20.    1
SYMBOL, SR =   101
SSINPT   =     1
S
SSOUTP   =     7
S1        SC1        SSAV       CAV        GAN        PHS        CPU
SSTABS   =     0
SYMBOL, SS =   101
STINPT   =     2
S        1 STR
STOUTP   =     4
MN        MAX        MIN        SIG
STTABS   =     0
SYMBOL, ST =   101
SUINPT   =     4
F    3   1 T    3   1 F    3   3 T    3   3
SUOUTP   =     2
F    3   2 T    3   2
SUTABS   =     0
SYMBOL, SU =   101
SVINPT   =     6
F    3   1 T    3   1 F    3   3 T    3   3 F    3   4 T    3   4
SVOUTP   =     2
F    3   2 T    3   2
SVTABS   =     0
SYMBOL, SV =   101
SWINPT   =     6
S    N   1 S    N   3 SW1        TC1        TC2        N     1 1
SWOUTP   =     1
S    N   2
SWTABS   =     0
SYMBOL, SW =   101
MODES = SW
SXINPT   =     6
```

```
S     N   1 S     N   3 S     N   5 S     N   6 SW1         TC1         TC2         N     1
SXOUTP   =     2
S     N   2 S     N   4
SXTABS   =     0
SYMBOL, SX =    101
MODES = SX
SYINPT   =    10
S     N   1 S     N   3 S     N   5 S     N   6 S     N   7 S     N   9 SW1         TC1
TC2           N     1 1
SYOUTP   =     3
S     N   2 S     N   4 S     N   8
SYTABS   =     0
SYMBOL, SY =    101
MODES = SY
TAINPT   =     0
TAOUTP   =     4
S          2 S          3 S          4 S          5
TATABS   =     4
A2T          39.    1
B2T          39.    1
C2T          39.    1
D2T          39.    1
SYMBOL, TA =    101
TBINPT   =     0
TBOUTP   =     2
S          2 S          3
TBTABS   =     2
A2T          39.    1
B2T          39.    1
SYMBOL, TB =    101
TDINPT   =     4
T     3     IXX          IYY          IZZ
TDOUTP   =     3
W     3     SEA     3     SWD     3
TDTABS   =     0
SYMBOL, TD =    101
TFINPT   =     6
S     N   1 Z0    N     Z1    N     P0    N     P1    N     N     1 1
TFOUTP   =     2
X1    N     SS    N    2S
TFTABS   =     0
SYMBOL, TF =    101
MODES = TF
TGINPT   =     3
TH          GAM   3     X     3
TGOUTP   =     2
F     3     T     3
TGTABS   =     0
SYMBOL, TG =    101
TRINPT   =    11
AL          ALD         GTA         CH          CN          CHP         CHG         CMF
WH          WHP         WMF
TROUTP   =     1
WO
TRTABS   =     0
SYMBOL, TR =    101
TSINPT   =     2
T     3   1 T     3   3
```

```
TSOUTP   =     1
T    3   2
TSTABS   =     0
SYMBOL, TS =    101
TTINPT   =     3
T    3   1 T    3   3 T     3   4
TTOUTP   =     1
T    3   2
TTTABS   =     0
SYMBOL, TT =    101
USINPT   =    10
LMP   M M   MS1   M M   WRK   M     STF   M M   THR   3 1   LMN   3 1   GNF   N 1   DLM   2
N    1 1   M     1 1
USOUTP   =    10
UVW  3    SPGR  3    SSD1  2    SSD2  2    SDLD  2    SFXD  N    SSL1  2    SSL2  2
DLT  2    SFLX  N    S
USTABS   =     0
SYMBOL, US =    101
MODES = US
UTINPT   =    40
PGR  3     MS1          MS2          LS1          LS2          SP1  3      SP2  3      ME
LE         EP    3      MSS          IXX          IYY          IZZ         IXY         IXZ
IYZ        IYE          IZE          IYH          IZH          MM   N      PS1  2 N    PS2  2
PE   2 N   PEP  2 N     WP1          WT1          WP2          WT2         WFX  N      WEP
WET        ZS1          ZS2          ZFX   N      ZEP          ZET         N    1 1    M    1
UTOUTP   =     5
DMP   M M   MS1   M M   STF   M M   MAS   M M   LQW   M
UTTABS   =     0
SYMBOL, UT =    101
MODES = UT
VAINPT   =     9
T    3     IN    3 3    LA       1 LO       1 TI       1 DA       1 ROL        PIT
YAW
VAOUTP   =     8
W    3    SQ     4     SA     3 3  EA     3     LA       2 LO       2 TI       2 DA
VATABS   =     0
SYMBOL, VA =    101
V6INPT   =    15
VPF        VL           ED           EQ           VRE          G1          G2          K
T1         T2           T3           T4           CEX          EB          G3
V6OUTP   =     7
E2        SE4          SE5          SVO          SEL          E1          E3
V6TABS   =     0
SYMBOL, V6 =    101
WBINPT   =    17
AB         SW           SX    3      SM    3      SP    3      W        1 X     3   1 BM    3
BP    3   1 W        2 X    3   2 BM   3   2 BP    3   2 W        3 X     3   3 BM    3
BP    3   3
WBOUTP   =     4
CW         CCG  3      CMI  3      CPI  3
WBTABS   =     0
SYMBOL, WB =    101
WMINPT   =    10
NU         NV           NW           NP           SLH          SLV         VS        1 SIH
SIV        B
WMOUTP   =    11
CW        SVW          SVX          SWW          SWX          SPW          SQX          SQW
RA        SRW          VS       2
```

```
WMTABS   =     0
SYMBOL, WM =    101
XPINPT   =     2
W    3   1 TRN   3 3
XPOUTP   =     1
W    3   2
XPTABS   =     0
SYMBOL, XP =    101
XTINPT   =     2
T    3   1 TRN   3 3
XTOUTP   =     1
T    3   2
XTTABS   =     0
SYMBOL, XT =    101
```

APPENDIX J

EASIEST F-4E MANEUVERING COEFFICIENTS

This appendix contains a listing of the EASIEST F-4E airplane maneuvering coefficients formatted for the EASIEST airplane modeled by component AE.

```
  42
  66   11   11   67   11   60   58   68   11   11   67   31   40   40   40   40   40   58
  67   58   49   67   58   49   49   40   49    8    9    8    6    4    7    5    3    5
   5    4    5    4    4    7
   1    0    1    2
(7E10.0)
2.        30.        28.
   1    3    1    4    LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
2.        1.
   1    5    1   68    CZO(A,M)   Z AXIS BIAS COEFFICIENT FOR TRIM
 (7F10.0/F10.0 )
-.22      .02        .29       .55       .8        .975      1.07      CL M=.2
1.235                                                                  CL M=.2
-.22      .02        .29       .55       .8        .975      1.07      CL M=.6
1.235                                                                  CL M=.6
-.23      .03        .29       .55       .81       .975      1.08      CL M=.7
1.21                                                                   CL M=.7
-.23      .03        .29       .57       .825      .975      1.1       CL M=.8
1.21                                                                   CL M=.8
-.26      .03        .32       .62       .85       .995      1.125     CL M=.9
1.235                                                                  CL M=.9
-.26      .04        .34       .65       .90       1.12      1.16      CL M=1.
1.2                                                                    CL M=1.
-.25      0.         .235      .445      .655      .85       1.01      CL M=1.
1.17                                                                   CL M=1.
-.2       -.035      .125      .295      .455      .605      .76       CL M=2.
.91                                                                    CL M=2.
   2    0    2    1
(7E10.0)
1.        29.
   2    2    2    2    LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.
   2    3    2   11    CZAD(M)   VARIATION OF CZO WITH ALPHA DOT
(7E10.0/2E10.0)
2.25      2.45       2.45      2.35      1.85      -1.25     -4.0      CLAD
-1.1      -.65                                                         CLAD
   3    0    3    1
(7E10.0)
1.        29.
   3    2    3    2   LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.
   3    3    3   11    CZQ(M)   VARIATION OF CZO WITH PITCH RATE
(7E10.0/2E10.0)
3.9       3.6        3.7       3.85      4.1       4.5       5.32      CLQ
2.9       1.3                                                          CLQ
   4    0    4    2
(7E10.0)
2.        29.        33.
   4    3    4    4   LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
   4    5    4   67    CZDE(M,A)   VARIATION OF CZO WITH ELEVATOR POSITION
(7F10.0/2F10.0)
0.0113    0.0107     0.010     0.0090    0.0082    0.0076    0.0078    CLDS A=
.00545    .0035                                                        CLDS A=
```

```
0.0109    0.0102    0.0095    0.0087    0.0080    0.00765   0.0077    CLDS A=
.00545    .0035                                                       CLDS A=
0.0104    0.00995   0.0093    0.0085    0.00779   0.0075    0.0075    CLDS A=
.00545    .0035                                                       CLDS A=
0.0109    0.0102    0.0096    0.0087    0.00782   0.0073    0.0074    CLDS A1
.0054     .0035                                                       CLDS A1
0.0105    0.00965   0.0091    0.0082    0.0074    0.00715   0.0073    CLDS A1
.00528    .0035                                                       CLDS A1
0.0092    0.0081    0.0073    0.00645   0.0058    0.0058    0.0066    CLDS A2
.0043     .0035                                                       CLDS A2
0.0067    0.0058    0.0053    0.0050    0.00475   0.0047    0.0056    CLDS A2
.0037     .0035                                                       CLDS A2
   5    0    5    1
(7E10.0)
1.        29.
   5    2    5    2  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.
   5    3    5   11   CZDA(M)   VARIATION OF CZO WITH AILERON POSITION
(7E10.0/2E10.0)
.0025     .0022     .002      .00177    .00162    .00152    .00135    CLDAE
.00035    .00028                                                      CLDAE
   6    0    6    2
(7E10.0)
2.        42.       28.
   6    3    6    4   LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
4.        1.
   6    5    6   60   CXO(CL,M)   X AXIS BIAS COEFFICIENT FOR TRIM
(7E10.0)
.026      .0315     .0485     .077      .13       .18       .26       CD M=.2
.026      .0315     .0485     .077      .13       .18       .26       CD M=.6
.025      .0305     .046      .0745     .13       .18       .27       CD M=.7
.024      .0295     .045      .075      .136      .2        .29       CD M=.8
.027      .033      .0485     .083      .155      .23       .33       CD M=.9
.047      .054      .076      .118      .195      .253      .345      CD M=1.
.0465     .063      .12       .241      .445      .547      .649      CD M=1.
.0445     .065      .133      .28       .49       .595      .7        CD M=2.
   7    0    7    2
(7E10.0)
2.        29.       31.
   7    3    7    4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
   7    5    7   58   CXDA(M,A)   VARIATION OF CXO WITH AILERON POSITION
(7E10.0/2E10.0)
0.00062   0.00064   0.00064   0.00065   0.00077   0.00118   0.00090   CDDAE A
.00048    .00044                                                      CDDA A
0.00076   0.00078   0.00078   0.00078   0.00088   0.00132   0.00107   CDDAE A
.0056     .00052                                                      CDDA A
0.00102   0.00104   0.00104   0.00105   0.00112   0.00156   0.00122   CDDAE A
.00062    .00058                                                      CDDA A1
0.00111   0.00114   0.00114   0.00115   0.00122   0.00170   0.00138   CDDAE A
.00067    .00063                                                      CDDA A1
0.00098   0.00100   0.00100   0.00100   0.00108   0.00146   0.00148   CDDAE A
.00078    .00074                                                      CDDA A2
0.00098   0.00100   0.00100   0.00100   0.00108   0.00146   0.00148   CDDAE A
0.00078   0.00074                                                     CDDA A2
```

```
   8   0   8   2
(7E10.0)
2.        30.       28.
   8   3   8   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
2.        1.
   8   5   8  68   CMO(A,M)   BIAS PITCHING MOMENT COEFFICIENT FOR TRIM
 (7F10.0/F10.0 )
.003      -.012     -.027     -.042     -.055     -.058     -.076     CM M=.2
-.117                                                                 CM M=.2
.004      -.011     -.026     -.039     -.047     -.047     -.074     CM M=.6
-.118                                                                 CM M=.6
.007      -.008     -.024     -.039     -.042     -.045     -.085     CM M=.7
-.115                                                                 CM M=.7
.012      -.006     -.024     -.04      -.042     -.051     -.101     CM M=.8
-.126                                                                 CM M=.8
.017      -.011     -.039     -.06      -.066     -.079     -.116     CM M=.9
-.138                                                                 CM M==9
.053      -.014     -.08      -.14      -.193     -.209     -.209     CM M=1.
-.209                                                                 CM M=1.
.053      -.006     -.064     -.116     -.161     -.202     -.245     CM M=1=
-.286                                                                 CM M=1.
.051      .008      -.037     -.077     -.112     -.146     -.182     CM M=2.
-.218                                                                 CM M=2=
   9   0   9   1
(7E10.0)
1.        29.
   9   2   9   2  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.
   9   3   9  11  CMAD(M)   VARIATION OF CMO WITH ALPHA DOT
(7E10.0/2E10.0)
-1.3      -1.25     -1.25     -1.37     -1.45     -1.35     -0.9      CMAD
.22       .8                                                          CMAD
  10   0  10   1
(7E10.0)
1.        29.
  10   2  10   2  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.
  10   3  10  11   CMQ(M)   VARIATION OF CMO WITH PITCH RATE
(7E10.0/2E10.0)
-3.7      -3.45     -3.33     -3.17     -3.15     -3.5      -5.15     CMQ
-3.02     -1.88                                                       CMQ
  11   0  11   2
(7E10.0)
2.        29.       33.
  11   3  11   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  11   5  11  67   CMDE(M,A)   VARIATION OF CMO WITH ELEVATOR POSITION
 (7F10.0/2F10.0)
-0.017    -0.01558  -0.01445  -0.0131   -0.012    -0.0121   -0.0121   CMDS A=
-.0084    -.0055                                                      CMDS A=
-0.016    -0.01495  -0.01395  -0.0127   -0.01175  -0.0118   -0.0120   CMDS A=
-.0084    -.0055                                                      CMDS A=
-0.0156   -0.01455  -0.01360  -0.01235  -0.01145  -0.0115   -0.01178  CMDS A=
-.0884    -.0055                                                      CMDS A=
```

```
-0.016    -0.01495  -0.01395  -0.0127   -0.01150  -0.01125  -0.0116   CMDS A1
-.0083    -.0055                                                      CMDS A1
-0.0154   -0.01445  -0.01340  -0.01185  -0.01080  -0.01115  -0.0114   CMDS A1
-.0062    -.0055                                                      CMDS A1
-0.01365  -0.0119   -0.01565  -0.0095   -0.0065   -0.0091   -0.0102   CMDS A2
-.0079    -.0055                                                      CMDS A2
-0.0099   -0.0084   -0.00775  -0.0071   -0.00675  -0.0073   -0.0087   CMDS A2
-.0069    -.0055                                                      CMDS A2
  12   0  12   2
(7E10.0)
2.        29.       35.
  12   3  12   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  12   5  12  31  CMDA(M,A)   VARIATION OF CMO WITH AILERON POSITION
(7E10.0/2E10.0)
-0.00054  -0.00052  -0.00052  -0.00052  -0.00053  -0.00069  -0.00096  CMDAF A
-.0004    -.00035                                                     CMDA A
-0.00040  -0.00040  -0.00040  -0.00044  -0.00048  -0.00060  -0.00080  CMDAE A
-.00063   -.00062                                                     CMDA A1
-0.00032  -0.00032  -0.00035  -0.00038  -0.00042  -0.00051  -0.00049  CMDAE A
-0.0002   -0.0002                                                     CMDA A2
  13   0  13   2
(7E10.0)
2.        27.       32.
  13   3  13   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  13   5  13  40  CYB(M,A)   VARIATION OF CY WITH BETA
(7E10.0/2E10.0)
-.0115    -.0117    -.0118    -.0121    -.0124    -.0129    -.0143    CYB A=0
-.0144    -.0118                                                      CYB A=0
-.0113    -.0113    -.0114    -.0116    -.0119    -.0125    -.0134    CYB A=8
-.013     -.0104                                                      CYB A=8
-.0106    -.0106    -.0107    -.0108    -.0111    -.0114    -.0119    CYB A=1
-.012     -.0091                                                      CYB A=1
-.0096    -.0098    -.0096    -.0095    -.0094    -.0097    -.0102    CYB A=2
-.0108    -.0108                                                      CYB A=2
  14   0  14   2
(7E10.0)
2.        29.       40.
  14   3  14   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  14   5  14  40  CPY(M,A)   VARIATION OF CY WITH ROLL RATE
(7E10.0/2E10.0)
-0.06     -0.065    -0.065    -0.07     -0.06     -0.03     -0.03     CYP  A=
-.03      -.03                                                        CYP A=0
+0.23     +0.24     +0.23     +0.20     +0.15     +0.34     +0.42     CYP  A=
.21       .025                                                        CYP A=8
+0.235    +0.24     +0.21     +0.17     +0.11     +0.28     +0.60     CYP  A1
.32       .07                                                         CYP A=1
+0.285    +0.24     +0.21     +0.17     +0.11     +0.28     +0.60     CYP  A2
.32       .07                                                         CYP A=2
  15   0  15   2
(7E10.0)
2.        29.       36.
  15   3  15   4  LOCATION OF INDEPENDENT VARIABLES
```

```
(7E10.0)
1.        2.
  15   5  15  40  CYR(M,A)   VARIATION OF CY WITH YAW RATE
(7E10.0/2E10.0)
0.735     0.78      0.80      0.835     0.88      0.88      0.56      CYR  A=
.455      .31                                                         CYR A=0
0.785     0.80      0.82      0.855     0.90      0.915     0.61      CYR  A1
.477      .273                                                        CYR A=1
0.785     0.80      0.82      0.855     0.90      0.915     0.61      CYR  A1
.477      .273                                                        CYR A=1
0.650     0.66      0.67      0.70      0.735     0.75      0.49      CYR  A2
.433      .433                                                        CYR A=2
  16   0  16   2
(7E10.0)
2.        29.       41.
  16   3  16   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  16   5  16  40  CYDR(M,A)   VARIATION OF CY WITH RUDDER POSITION
(7E10.0/2E10.0)
.00219    .0021     .00208    .00204    .00189    .00163    .00149    CYDR A=
.00126    .00097                                                      CYDR A=
.00219    .0021     .00206    .00204    .00189    .00163    .00149    CYDR A1
.00126    .00097                                                      CYDR A1
.00206    .00197    .00195    .00191    .00177    .00152    .0014     CYDR A2
.00117    .00088                                                      CYDR A2
.00186    .0018     .00178    .00174    .00161    .00139    .00128    CYDR A2
.00105    .00076                                                      CYDR A2
  17   0  17   2
(7E10.0)
2.        29.       32.
  17   3  17   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  17   5  17  40  CYDA(M,A)   VARIATION OF CY WITH AILERON POSITION
(7E10.0/2E10.0)
.000265   .000265   .000265   .000265   .000265   .000265   .000265   CYDA A=
.000145   .00007                                                      CYDA A=
.0002     .0002     .0002     .0002     .0002     .0002     .0002     CYDA A=
.000005   -.00006                                                     CYDA A=
-.000017  -.000017  -.000017  -.000017  -.000017  -.000017  -.000017  CYDA A1
-.00012   -.00018                                                     CYDA A1
0.        0.        0.        0.        0.        0.        0.        CYDA A2
-.000103  -.000163                                                    CYDA A2
  18   0  18   2
(7E10.0)
2.        29.       31.
  18   3  18   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  18   5  18  58  CLB(M,A)   VARIATION OF CL WITH BETA
(7E10.0/2E10.0)
-.00055   -.00054   -.00055   -.00062   -.00076   -.00087   -.00085   CLB A=0
-.00019   0.                                                          CLB A=0
-.00183   -.00192   -.00198   -.00205   -.00216   -.00225   -.0021    CLB A=8
-.00027   0.                                                          CLB A=6
-.00243   -.00247   -.00245   -.00243   -.0024    -.00235   -.0023    CLB A=1
-.00045   -.0004                                                      CLB A=1
```

```
-.0028    -.00262   -.00265   -.0027    -.00277   -.00286   -.0029    CLB A=1
-.00067   -.00067                                                     CLB A=1
-.0028    -.00312   -.00325   -.00338   -.00351   -.0036    -.00362   CLB A=2
-.0016    -.0016                                                      CLB A=2
-.0028    -.00399   -.00412   -.00423   -.00435   -.00447   -.00452   CLB A=2
-.0044    -.0044                                                      CLB A=2
  19   0  19   2
(7E10.0)
2.        29.       33.
  19   3  19   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  19   5  19  67  CLP(M,A)   VARIATION OF CL WITH ROLL RATE
(7E10.0/2E10.0)
-0.29     -0.287    -0.285    -0.26     -0.291    -0.338    -0.345    CLP  A=
-.266     -.214                                                       CLP A=0
-0.295    -0.295    -0.290    -0.29     -0.304    -0.338    -0.332    CLP  A=
-.253     -.207                                                       CLP A=4
-0.30     -0.304    -0.30     -0.30     -0.302    -0.324    -0.319    CLP  A=
-.247     -.203                                                       CLP A=8
-0.31     -0.310    -0.31     -0.306    -0.270    -0.263    -0.300    CLP  A1
-.24      -.198                                                       CLP A=1
-0.271    -0.270    -0.261    -0.245    -0.215    -0.218    -0.268    CLP  A1
-0.225    -.183                                                       CLP A=1
-0.248    -0.247    -0.233    -0.205    -0.173    -0.170    -0.228    CLP  A2
-.195     -.155                                                       CLP A=2
-0.265    -0.211    -0.20     -0.160    -0.130    -0.133    -0.188    CLP  A2
-.17      -.13                                                        CLP A=2
  20   0  20   2
(7E10.0)
2.        29.       31.
  20   3  20   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  20   5  20  58  CLR(M,A)   VARIATION OF CL WITH YAW RATE
(7E10.0/2E10.0)
.015      .02       .015      .015      .018      .03       .05       CLR A=0
.07       .031                                                        CLR A=0
.094      .1        .107      .115      .136      .14       .07       CLR A=8
.075      .038                                                        CLR A=8
.132      .148      .157      .168      .183      .14       .057      CLR A=1
.06       .031                                                        CLR A=1
.165      .188      .198      .21       .226      .14       .01       CLR A=1
.005      .018                                                        CLR A=1
.193      .22       .233      .247      .266      .14       -.025     CLR A=2
-.045     -.045                                                       CLR A=2
.2        .255      .27       .285      .305      .14       -.04      CLR A=2
-.065     -.065                                                       CLR A=2
  21   0  21   2
(7E10.0)
2.        29.       36.
  21   3  21   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  21   5  21  49  CLDR(M,A)   VARIATION OF CL WITH RUDDER POSITION
(7E10.0/2E10.0)
+0.000225 +0.000235 +0.000245 +0.000250 +0.000250 +0.000222 +0.000210 CLDR A=
.00014    .000105                                                     CLDR A=
```

```
+0.000030 +0.000035 +0.000035 +0.000035 +0.000035 +0.000035 +0.000030 CLDP  A=
0.        0.                                                          CLDR  A=
-0.000050 -0.000055 -0.000055 -0.000055 -0.000055 -0.000055 -0.000055 CLDR  A1
-.000065  -.000055                                                    CLDR  A1
-0.000120 -0.000120 -0.000130 -0.000130 -0.000130 -0.000120 -0.000110 CLDR  A1
-.0001    -.0001                                                      CLDR  A1
-0.000250 -0.000260 -0.000270 -0.000275 -0.000270 -0.000250 -0.000230 CLDR  A2
-.00015   -.00015                                                     CLDR  A2
  22    0   22    2
(7E10.0)
2.        29.       33.
  22    3   22    4   LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  22    5   22   67   CLDA(M,A)     VARIATION OF CL WITH AILERON POSITION
(7E10.0/2E10.0)
-.000745  -.00081   -.000825  -.000845  -.00088   -.00089   -.00078   CRDA  A=
-.000355  -.00023                                                     CRDA  A=
-.0008    -.000875  -.000905  -.00094   -.00097   -.000995  -.000865  CRDA  A=
-.00036   -.00023                                                     CRDA  A=
-.000375  -.000915  -.000945  -.00094   -.00087   -.00078   -.00073   CRDA  A=
-.00033   -.00022                                                     CRDA  A=
-.0008    -.00085   -.00081   -.00073   -.00055   -.00046   -.0005    CRDA  A1
-.0003    -.00021                                                     CRDA  A1
-.000645  -.0006    -.000545  -.00047   -.00041   -.000355  -.00039   CRDA  A1
-.00026   -.0002                                                      CRDA  A1
-.0004    -.00026   -.00022   -.00017   -.00016   -.000165  -.00023   CRDA  A2
-.000245  -.000245                                                    CRDA  A2
-.000305  -.000205  -.00018   -.00015   -.00011   -.00009   -.000095  CRDA  A2
-.000125  -.000125                                                    CRDA  A2
  23    0   23    2
(7E10.0)
2.        29.       31.
  23    3   23    4   LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  23    5   23   58   CNB(M,A)    VARIATION OF CN WITH BETA
(7E10.0/2E10.0)
.00192    .0018     .0018     .00184    .00198    .0024     .00307    CNB  A=0
.00246    .0012                                                       CNB  A=0
.00195    .00194    .00195    .00198    .00212    .0023     .00265    CNB  A=8
.00246    .00084                                                      CNB  A=8
.00202    .00205    .00207    .00219    .00224    .00235    .00257    CNB  A=1
.0029     .00087                                                      CNB  A=1
.00205    .00194    .00193    .00192    .00192    .00205    .0024     CNB  A=1
.00226    .00226                                                      CNB  A=1
.00213    .0015     .00153    .0016     .0017     .00187    .00212    CNB  A=2
.00377    .00377                                                      ENB  A=2
.00152    .00105    .00115    .0013     .00147    .00167    .00192    CNB  A=2
.00357    .00357                                                      CNB  A=2
  24    0   24    2
(7E10.0)
2.        29.       39.
  24    3   24    4   LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  24    5   24   49   CNP(M,A)    VARIATION OF CN WITH ROLL RATE
(7E10.0/2E10.0)
```

```
+0.005    +0.005    +0.005    +0.005    +0.005    +0.005    +0.005    CNP  A=
.005      .005                                                        CNP A=0
-0.035    -0.045    -0.040    -0.032    -0.032    -0.040    -0.055    CNP  A=
-.017     .005                                                        CNP A=4
-0.073    -0.058    -0.050    -0.042    -0.040    -0.10     -0.20     CNP  A=
-.042     -.003                                                       CNP A=8
-0.068    -0.068    -0.064    -0.055    -0.050    -0.075    -0.125    CNP  A1
-.07      -.01                                                        CNP A=1
-0.058    -0.068    -0.064    -0.055    -0.050    -0.075    -0.125    CNP  A2
-0.07     -.01                                                        CNP A=2
  25   0  25   2
(7E10.0)
2.        29.       37.
  25   3  25   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  25   5  25  49  CNR(M,A)   VARIATION OF CN WITH YAW RATE
(7E10.0/2E10.0)
-0.307    -0.32     -0.325    -0.34     -0.357    -0.374    -0.38     CNR  A=
-.332     -.205                                                       CNR A=0
-0.325    -0.341    -0.350    -0.365    -0.375    -0.380    -0.38     CNR  A1
-.31      -.18                                                        CNR A=1
-0.340    -0.370    -0.382    -0.395    -0.403    -0.403    -0.39     CNR  A1
-.31      -.18                                                        CNR A=1
-0.375    -0.40     -0.41     -0.42     -0.425    -0.410    -0.39     CNR  A2
-.31      -.18                                                        CNR A=2
-0.425    -0.435    -0.44     -0.445    -0.435    -0.410    -0.39     CNR  A2
-.31      -.18                                                        CNR A=2
  26   0  26   2
(7E10.0)
2.        29.       41.
  26   3  26   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  26   5  26  40  CNDR(M,A)   VARIATION OF CN WITH RUDDER POSITION
(7E10.0/2E10.0)
-.00131   -.0013    -.00126   -.00124   -.00118   -.00111   -.001     CNDR A=
-.00052   -.00031                                                     CNDR A=
-.00131   -.0013    -.00126   -.00124   -.00118   -.00111   -.001     CNDR A1
-.00052   -.00031                                                     CNDR A1
-.00122   -.00121   -.00118   -.00115   -.0011    -.00103   -.00093   CNDR A2
-.00045   -.00045                                                     CNDR A2
-.00111   -.00111   -.00108   -.00106   -.00101   -.00095   -.00086   CNDR A2
-.00036   -.00036                                                     CNDR A2
  27   0  27   2
(7E10.0)
2.        29.       34.
  27   3  27   4  LOCATION OF INDEPENDENT VARIABLES
(7E10.0)
1.        2.
  27   5  27  49  CNDA(M,A)   VARIATION OF CN WITH AILERON POSITION
(7E10.0/2E10.0)
-.000126  -.000058  -.000063  -.000073  -.000089  -.000108  -.000123  CNDA A=
-.000045  -.000024                                                    CNDA A=
.00002    .000075   .000077   .000076   .000074   .000065   .000048   CNDA A=
.00006    .000075                                                     CNDA A=
.00017    .000208   .000208   .000202   .000192   .00018    .000172   CNDA A1
.000212   .000223                                                     CNDA A1
```

```
.000178    .000245    .000237    .000222    .000205    .000195    .00019     CNDA A2
.000192    .000192                                                          CNDA A2
.00013     .00025     .000227    .00018     .000133    .00011     .000097    CNDA A2
.00009     .00009                                                           CNDA A2
  28    0   28    0  NM1
(7E10.0)
8.
  28    1   28    8   MACH NUMBER TABLE 1
(7E10.0/E10.0)
0.2        0.5        0.7        0.8        0.9        1.1        1.6        MACH 1
2.1                                                                         MACH 1
  29    0   29    0   NM2
(7E10.0)
9.
  29    1   29    9   MACH NUMBER TABLE 2
(7E10.0/2E10.0)
0.2        0.6        0.7        0.8        0.9        1.0        1.1        MACH 2
1.6        2.1                                                              MACH 2
  30    0   30    0   NA1
(7E10.0)
8.
  30    1   30    8   ALPHA TABLE 1
(7E10.0/E10.0)
-4.        0.         4.         8.         12.        16.        20.        ALPHA1
24.                                                                         ALPHA1
  31    0   31    0  NA2
(7E10.0)
6.
  31    1   31    6   ALPHA TABLE 2
(7E10.0)
0.         8.         12.        16.        20.        24.                   ALPHA2
  32    0   32    0  NA3
(7E10.0)
4.
  32    1   32    4   ALPHA TABLE 3
(7E10.0)
0.         8.         16.        24.                                          ALPHA3
  33    0   33    0   NA4
(7E10.0)
7.
  33    1   33    7   ALPHA TABLE 4
(7E10.0)
0.         4.         8.         12.        16.        20.        24.        ALPHA4
  34    0   34    0   NA5
(7E10.0)
5.
  34    1   34    5   ALPHA TABLE 5
(7E10.0)
0.         8.         16.        20.        24.                              ALPHA5
  35    0   35    0   NA6
(7E10.0)
3.
  35    1   35    3   ALPHA TABLE 6
(7E10.0)
0.         16.        24.                                                   ALPHA6
  36    0   36    0   NA7
(7E10.0)
5.
```

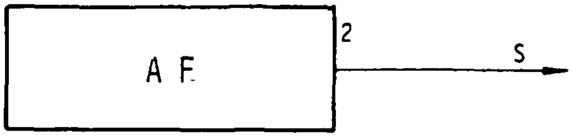
```
  36    1   36    5    ALPHA TABLE 7
(7E10.0)
0.         8.          12.         16.         24.                                                        ALPHA7
  37    0   37    0    NA8
(7E10.0)
5.
  37    1   37    5    ALPHA TABLE 8
(7E10.0)
0.         12.         16.         20.         24.                                                        ALPHA8
  38    0   38    0    NA9
(7E10.0)
4.
  38    1   38    4    ALPHA TABLE 9
(7E10.0)
0.         12.         16.         24.                                                                    ALPHA9
  39    0   39    0    NA10
(7E10.0)
5.
  39    1   39    5    ALPHA TABLE 10
(7E10.0)
0.         4.          8.          12.         24.                                                        ALPHA10
  40    0   40    0    NA11
(7E10.0)
4.
  40    1   40    4    ALPHA TABLE 11
(7E10.0)
0.         8.          12.         24.                                                                    ALPHA11
  41    0   41    0    NA12
(7E10.0)
4.
  41    1   41    4    ALPHA TABLE 12
(7E10.0)
0.         16.         20.         24.                                                                    ALPHA 1
  42    0   42    0    NCL
(7E10.0)
7.
  42    1   42    7    CL TABLE (INDEPENDENT)
(7E10.0)
0.         0.2         0.4         0.6         0.8         0.9         1.0         CLTABLE
-1
```

APPENDIX K

EASY5 INPUT/OUTPUT LISTS

This appendix contains input and output tables for the EASY5, (not EASIEST), standard components. Descriptive figures are also presented for the more complex components.

ANALYTIC FUNCTION GENERATOR **AF**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
COD		Specifies which analytic function is calculated. (See equations below for use of these inputs)	
C1			
C2			
C3			
C4			
C5			

COD = 1 $\quad S2 = C1 + C2 \cdot SIN(C3 \cdot t + C4)$

2 $\quad S2 = C1 + C2 \cdot COS(C3 \cdot t + C4)$

3 $\quad S2 = C1 + e^{-C5 \cdot t} \cdot (SIN(C3 \cdot t + C4))$

4 $\quad S2 = C1 + e^{-C5 \cdot t} \cdot (COS(C3 \cdot t + C4))$

5 $\quad S2 = C1 + C2 \cdot t$

6 $\quad S2 = C1 + C2 \cdot e$

where: t = TIME

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S	2	Output	

INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
U(3)		X, Y, Z BODY AXIS LINEAR VELOCITIES	FT/SEC
W(3)		X, Y, Z BODY AXIS ANGULAR RATES	DEG/SEC
ALT		ALTITUDE ABOVE SEA LEVEL	FT
EA(3)		PITCH, ROLL, YAW EARTH TO BODY AXIS ANGLES	DEG
ID	1	INDICATOR FUNCTION FOR AERO COMPONENTS 0 = BODY AXIS, DIMENSIONAL 1 = BODY AXIS, NON-DIMENSIONAL 2 = STABILITY AXIS, DIMENSIONAL 3 = STABILITY AXIS, NON-DIMENSIONAL	
VS		STEADY STATE (TRIM) AIRSPEED	FT/SEC
ALS*		STEADY STATE (TRIM) ANGLE OF ATTACK	DEG
S		REFERENCE AREA	FT^2
UW, VW, WW*, PW*		X, Y, Z BODY AXIS WIND VELOCITIES	FT/SEC
QW, RW*	1	X, Y, Z BODY AXIS WIND ANGULAR RATES	DEG/SEC

*DEFAULT VALUES = 0

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
UO(3)		X, Y, Z BODY AXIS VELOCITIES INCLUDING WIND	FT/SEC
WO(3)		X, Y, Z BODY AXIS ANGULAR RATES WITH WIND	DEG/SEC
ID	2	INDICATOR FUNCTION = ID1	
WW(3)	2	ANGULAR RATE DUE TO GUSTS	DEG/SEC
CAL, SAL		DIRECTION COSINES FOR STABILITY AND BODY AXES	
AL, ALP		ANGLE OF ATTACK IN BODY AND STABILITY AXES	DEG
VT		TRUE AIRSPEED	FT/SEC
BE		SIDESLIP ANGLE	DEG
WP, UP		Z & X STABILITY AXIS VELOCITIES (DIMENSIONAL) Z & X PERTURBATION VELOCITIES (NON-DIMEN.)	FT/SEC
EU(3)		X, Y, Z BODY AXIS ACCEL. TERMS FOR U, V, W SOLUTIONS	FT/SEC^2
SIG		STANDARD ATMOSPHERE AIR DENSITY RATIO	
QC		COMPRESSIBLE DYNAMIC PRESSURE	LBS/FT^2
QS		DYNAMIC PRESSURE TIMES REFERENCE AREA	LBS
MAC		MACH NUMBER	

VECTOR DEFINITIONS

AV

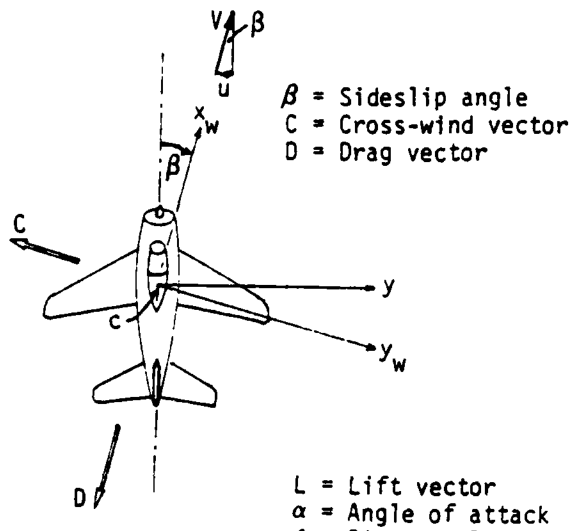
$$U(3) = \begin{pmatrix} U \\ V \\ W \end{pmatrix} \quad W(3) = \begin{pmatrix} P \\ Q \\ R \end{pmatrix} \quad EA(3) = \begin{pmatrix} PIT \\ ROL \\ YAW \end{pmatrix} \quad UO(3) = \begin{pmatrix} UO \\ VO \\ WO \end{pmatrix}$$

$$WO(3) = \begin{pmatrix} PO \\ QO \\ RO \end{pmatrix} \quad WW(3) = \begin{pmatrix} 0 \\ QW \\ RW \end{pmatrix} \quad EU(3) = \begin{pmatrix} EU \\ EV \\ EW \end{pmatrix}$$

AERODYNAMIC VARIABLE EQUATIONS

$$CAL = \begin{cases} COS(ALS) & ID = 2,3 \\ 1 & ID = 0,1 \end{cases}$$

$$SAL = \begin{cases} SIN(ALS) & ID = 2,3 \\ 0 & ID = 0,1 \end{cases}$$

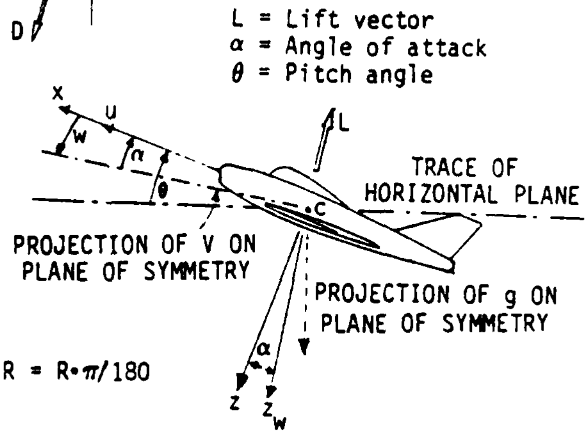


$$UO = U\text{-}UW$$
$$VO = V\text{-}VW$$
$$WO = W\text{-}WW$$

$$PO = (P+PW)\cdot CAL+(R+RW)\cdot SAL$$
$$QP = Q+QW$$
$$RO = (R+RW)\cdot CAL-(P+PW)\cdot SAL$$

$$AL = TAN^{-1}(WO/UO)$$
$$ALP = AL\text{-}ALS$$
$$VT = (UO^2+VO^2+WO^2)^{1/2}$$
$$BE = SIN^{-1}(VO/VT)$$
$$WP = WO\cdot CAL\text{-}UO\cdot SAL$$

$$UP = \begin{cases} UO\cdot CAL+WO\cdot SAL & ID = 0,2 \\ UO\text{-}VS\cdot COS(ALS)/VS & ID = 1 \\ UO\cdot CAL+WO\cdot SAL\text{-}VS)/VS & ID = 3 \end{cases}$$



$$EU = -Q\cdot W+R\cdot V - G\cdot SIN(PIT)$$
$$EV = -R\cdot U+P\cdot W + G\cdot COS(PIT)\cdot SIN(ROL)$$
$$EW = -P\cdot V+Q\cdot U + G\cdot COS(PIT)\cdot COS(ROL)$$

where $P = P\cdot\pi/180$, $Q = Q\cdot\pi/180$, $R = R\cdot\pi/180$

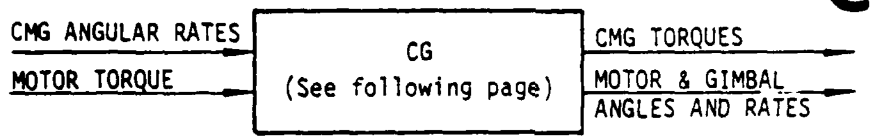
SIG = SIG(ALT) and A = A(ALT) obtained by table lookup

$$DPS = \tfrac{1}{2}\, PO\cdot SIG\cdot(VT)^2$$
$$QS = DPS\cdot S$$
$$MAC = VT/A$$

$$QC = \begin{cases} DPS\cdot(1+(1+(1+MAC^2/40)\ MAC^2/10)\ MAC^2/4) & MAC \le 1 \\ DPS\cdot(1.839 - .772/MAC^2 + .164/MAC^4 + .035/MAC^6) & MAC > 1 \end{cases}$$

CONTROL MOMENT GYRO

CG



INPUT

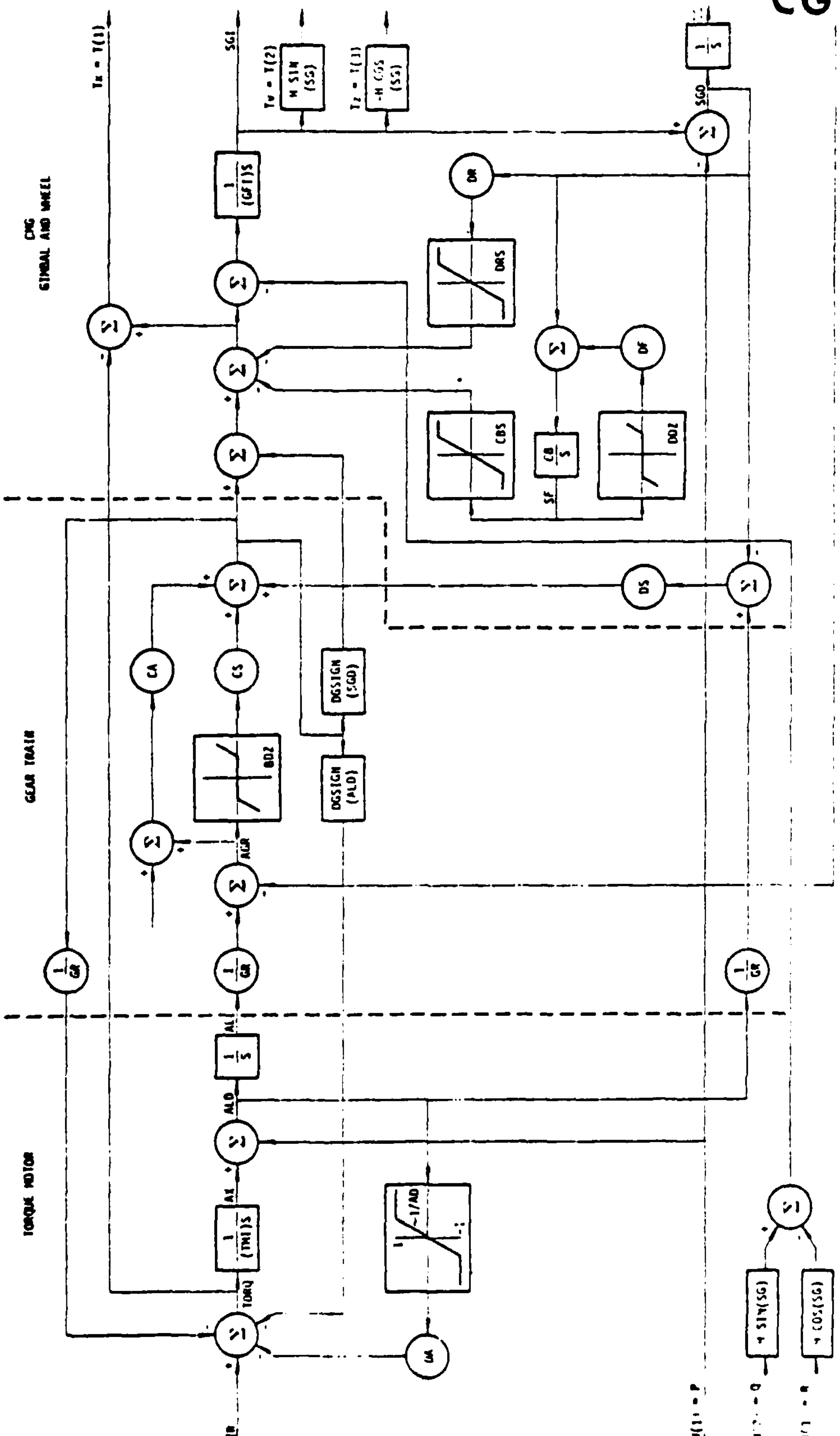
PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
W(3)		CMG Angular Rates; P, Q, R	rad/sec
TR		Motor Torque	ft-lbs
TMI		Torque Motor Inertia	slug-ft^2
AD		Torque Motor Rate Damping Limit	rad/sec
DA		Torque Motor Damping	ft-lb/rad/sec
GR		Gear Ratio	-
BDZ		Gear Backlash Deadzone	rad
CS		Gear Train Compliance	ft-lb/rad/sec
CA		Preload Spring Compliance	ft-lb/rad/sec
PLD		Preload Deadzone	rad
DG		Damping	ft-lb/rad
GFI		Gimbal Inertia	slug-ft^2
DR		Gimbal Damping Coefficient	ft-lb/rad/sec
DRS		Gimbal Damping Saturation Limit	ft-lbs
CB		Gimbal Friction Spring Term	ft-lb/rad/sec
CBS		Gimbal Friction Compliance Limit	ft-lbs
DDZ		Gimbal Damping Deadzone	rad
DF		Gimbal Friction Equivalent Spring	-
DS		Gimbal Viscous Friction	ft-lb/rad/sec
H		Angular Momentum	ft-lb-sec

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
*AL		Torque Motor Angle	rad
ALD		Torque Motor Rate	rad/sec
*AX		Torque Motor Intermediate State	rad/sec
*SG		Relative Gimbal Angle	rad
SGD		Relative Gimbal Angle Rate	rad/sec
*SGI		Inertial Gimbal Angle	rad
*SF		Gimbal Friction Spring Term	-
T(3)		CMG X, Y, Z Axis Torques	ft-lbs

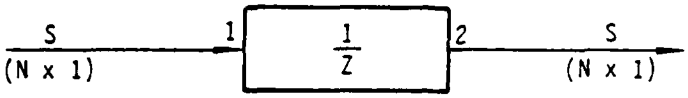
*These outputs are states

CG



DISCRETE DELAY

DE



INPUT

PHYCICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input quantity	
TAU		Sample period	seconds

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	Delay output (Delay state)	

EQUATIONS:

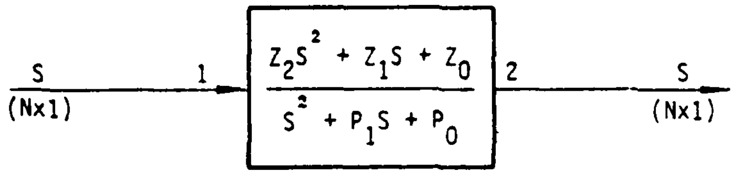
$$S2(N) = Z^{-1}\left[S1(N)\right]$$

$$Z^{-1}\left[\ \ \right] = \text{Discrete delay operator of TAU seconds}$$

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

DIGITAL FILTER

DF



INPUTS

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input Quantity	
Z0(N)		Numerator coefficient (S-plane)	
Z1(N)		Numerator coefficient (S-plane)	
Z2(N)		Numerator coefficient (S-plane)	
P0(N)		Denominator coefficient (S-plane)	
P1(N)		Denominator coefficient (S-plane)	
TAU		Sample period	sec

OUTPUTS

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	*Output quantity (Sample)*	
D1(N)		Intermediate output (Delay)	
D2(N)		Intermediate output (Delay)	

EQUATIONS:

$$D2 = Z^{-1}\ A2 \cdot S1 - B2 \cdot S2$$

$$D1 = Z^{-1}\ D2 + A1 \cdot S1 - B1 \cdot S2$$

$$S2 = A0 \cdot S1 + D1$$

$$Z^{-1}\Big[\ \Big] \equiv \text{discrete delay operator}$$

A0 ∿ A2 and B0 + B1 are related to S-plane coefficients by applying prewarping and bilinear transformation;

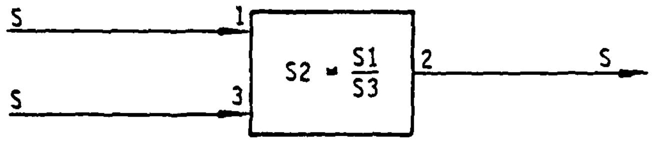
$$\frac{W_i}{TAU\left(\frac{W_i\tau}{2}\right)}\left(\frac{1-\Delta}{1+\Delta}\right)$$

to each of the singularities, W_i, of the numerator and denominator.

Note: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

DI

DIVIDER



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Numerator	
S(N)	3	Denominator	

OUTPUT

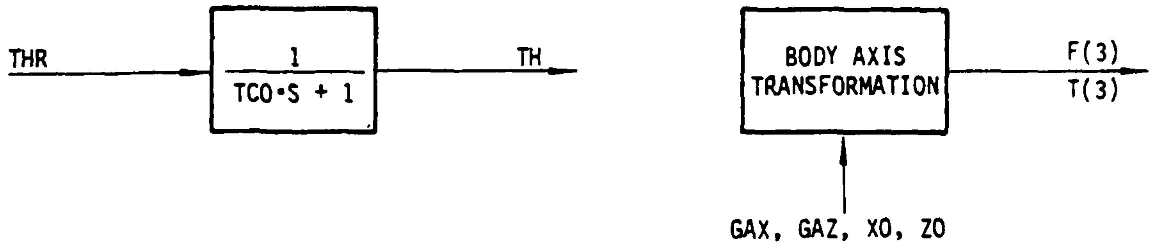
PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	Quotient	

EQUATIONS:

$$S2(N) = \frac{S1(N)}{S3(N)}$$

NOTE: N may be specified at Model Generation time, to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

FIRST ORDER LAG ENGINE MODEL **EN**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
TCO		Engine time constant	sec
THR		Required thrust level	lbs
GAX, GAZ		X, Z body axis direction cosines	
XO, ZO		X, Z thrust location components	ft

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
TH		Thrust output - state	lbs
F(3)		X, Y, Z body axis forces	lbs
T(3)		Axis torques (pitching moment)	ft-lbs

EQUATIONS

$\dot{TH} = (THR - TH)/TCO$

$F(1) = TH \cdot GAX$

$F(2) = 0$

$F(3) = TH \cdot GAZ$

$T(1) = 0$

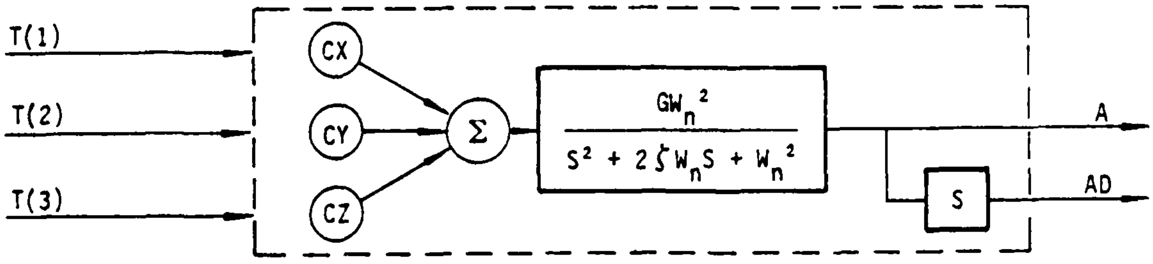
$T(2) = (ZO \cdot FX - X) \cdot FZ$

$T(3) = 0$

*TCO = Yields

TH = THR

TORQUES-TO-FLEXIBLE MODE AMPLITUDE AND RATE **FM**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
T(3)		X, Y, Z body axis torques	ft-lbs
GAI		Mode gain, G	rad/ft-lb-sec
DMP		Mode damping, ζ	
WN		Mode natural frequency, W_n	rad/sec
CX(3)		X, Y, Z body axis coefficients to convert	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
*A		Mode amplitude - state	rad
*AD		Mode rate - state	rad/sec

EQUATIONS OF MOTION:

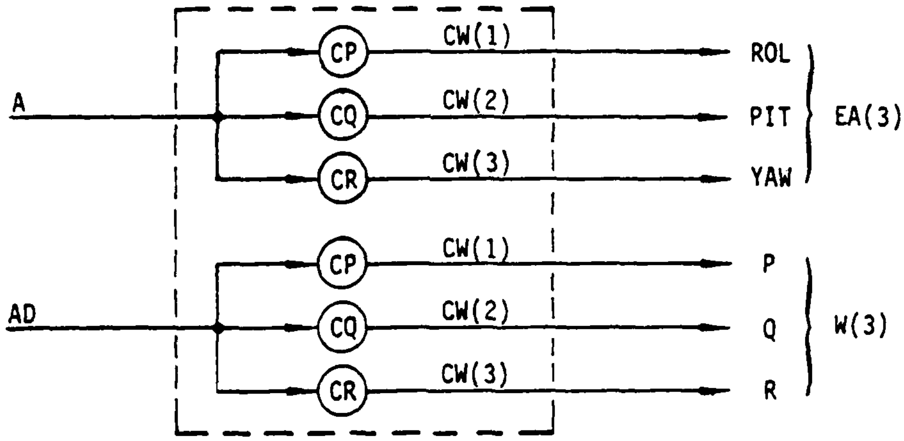
$\dot{AD}$ = ((GAI•(CX(1)•T(1) + CX(2)•T(2) + CX(3)•T(3)) - A)•WN - 2•DMP•AD)•WN

$\dot{A}$ = AD

NOTE:

This component is used with FP to produce angular rates due to flexible structure.

FLEXIBLE MODE AMPLITUDE-TO-DEFLECTIONS AND RATES **FP**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
A		Mode amplitude	rad
AD		Mode rate	rad/sec
CW(3)		X, Y, Z body axis coefficients to convert mode amplitude to body axis rates	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
EA(3)		X, Y, Z body axis angular deflections	rad
W(3)		X, Y, Z body axis rates	rad/sec

EQUATIONS:

ROL = CP•A
PIT = CQ•A
YAW = CR•A
P = CP•AD
Q = CQ•AD
R = CR•AD

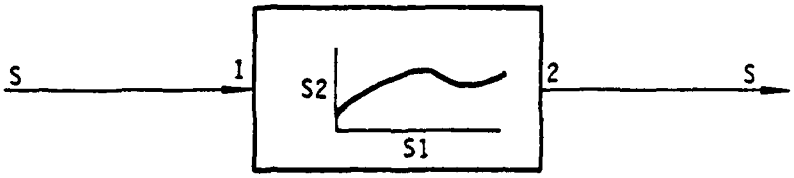
VECTOR DEFINITIONS:

$$CW(3) = \begin{pmatrix} CP \\ CQ \\ CR \end{pmatrix} \quad EA(3) = \begin{pmatrix} ROL \\ PIT \\ YAW \end{pmatrix} \quad W(3) = \begin{pmatrix} P \\ Q \\ R \end{pmatrix}$$

NOTE: This component is used with FM to produce angular deflections and rates due to flexible structure.

FUNCTION GENERATOR

FU



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S	1	Input quantity	
AN		Degree of interpolation (AN < 0 prevents extrapolation)	
FTA		Tabular values of function	

OUTPUT

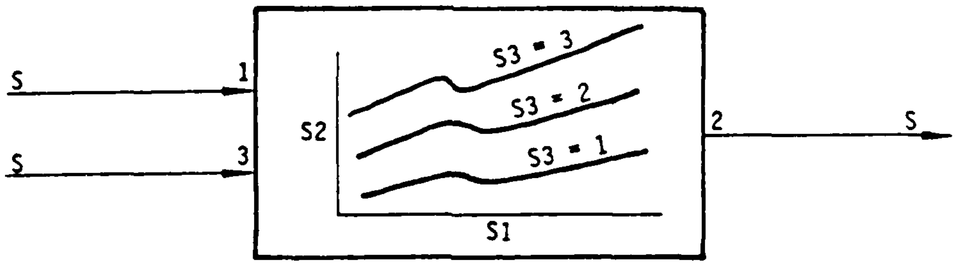
PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S	2	Output quantity	

EQUATION:

$$S2 = FTA(S1)$$

NOTE: A maximum of 18 points is allowed in the table

TWO-DIMENSIONAL FUNCTION **FV**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S	1	Input quantity	
S	3	Input quantity	
AN		Degree of interpolation for S1*	
BN		Degree of interpolation for S3*	
FTA		Table of functional relationships	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S	2	Output quantity	

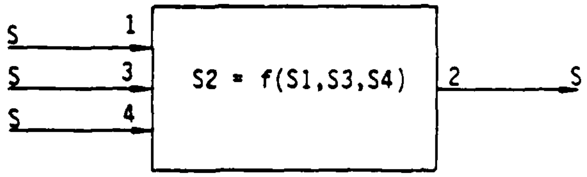
EQUATION:

$$S2 = FTA(S1, S3)$$

* A negative value for AN or BN prevents extrapolation beyond the table boundaries

FW

THREE-DIMENSIONAL FUNCTION



INPUT

PHYSICAL QUANTITY NAME	PORT	DESCRIPTION	UNITS
S	1	INPUT QUANTITY	
S	3	INPUT QUANTITY	
S	4	INPUT QUANTITY	
ANX		DEGREE OF INTERPOLATION FOR S1*	
ANY		DEGREE OF INTERPOLATION FOR S3*	
ANZ		DEGREE OF INTERPOLATION FOR S4*	
FTA		TABLE OF FUNCTIONAL RELATIONSHIPS	

OUTPUT

PHYSICAL QUANTITY NAME	PORT	DESCRIPTION	UNITS
S	2	OUTPUT QUANTITY	

EQUATION: S2 = FTS(S1,S3,S4)

* A NEGATIVE VALUE PREVENTS EXTRAPOLATION BEYOND THE TABLE BOUNDARIES.

FEEDER AND CIRCUIT BREAKER

F2

INPUT

PHYSICAL QUANTITY NAME	FIGURE 3.0-1 NAME	DESCRIPTION	UNITS
EDI	e_{dl}	Load Voltage, D-Axis Component	p.u.
EQI	e_{ql}	Load Voltage, Q-Axis Component	p.u.
ADI	i_d	Generator Current, D-Axis Component	p.u.
AQI	i_q	Generator Current, Q-Axis Component	p.u.
AB	I_B	Base Value of Current, Peak	amps
RNL	R_{NL}	No-Load Shunt Resistance Default Value = 50.0	p.u.
RS1	R_{s1}	Simulated Breaker Open Circuit Resistance	p.u.
RW	R_W	Feeder Resistance	p.u.
XW	X_W	Feeder Resistance	p.u.
XC	X_c	No-Load Shunt Capacitive Reactance Default Value = 50.0	p.u.
WO	ω_0	Base Frequency (WO = W_{zero})	rads/sec

OUTPUT

PHYSICAL QUANTITY NAME	FIGURE 3.0-1 NAME	DESCRIPTION	UNITS
*EDO	e_d	Generator Terminal Voltage, D-Axis Component	p.u.
*EQO	e_q	Generator Terminal Voltage, Q-Axis Component	p.u.
*ADO	i_{dl}	Load Current, D-Axis Component	p.u.
*AQO	i_{ql}	Load current, Q-Axis Component	p.u.
ARO	I_P	Real Current	amps
AIO	I_Q	Imaginary Current	amps
AT		Total Line Current, RMS	amps
PF		Power Factor	
RTF		Intermediate Quantity	p.u.
PHI	δ_L	Load Voltage D-Q Angle	radians

* This output quantity is a state.

F2

EQUATIONS:

```
RTF = RS1+RW
PHI = ATAN•(EDI/EQI)
ARO = (AB/1.4142)•(1/SQRT(EDI•EDI+EQI•EQI))•(EDI•ADO+EQI•AQO)
AIO = (AB/1.4142)•(1/SQRT(EDI•EDI+EQI•EQI))•(ADO•EQI-EDI•AQO)
 AT = SQRT(ARO•ARO+AIO•AIO)
 PF = COS(ATAN(AIO/ARO))
EDO = (WO/RNL)•(-EDO•XC+EQO•RNL+ADI•XC•RNL-ADO•XC•RNL)
EQO = (WO/RNL)•(-EQO•XC-EDO•RNL-AQO•XC•RNL+AQI•XC•RNL)
ADO = (WO/XW)•(-ADO•RTF+AQO•XW+EDO-EDI)
AQO = (WO/XW)•(-AQO•RTF-ADO•XW-EQI+EQO)
```

GENERATOR - EXCITER

G8

OUTPUT

PHYSICAL QUANTITY NAME	FIGURE 3.0-2 NAME	DESCRIPTION	UNITS
A1	i_d	Generator Current, D-Axis Component	p.u.
A2	i_q	Generator Current, Q-Axis Component	p.u.
A3	i_f	Generator Field Current	p.u.
*A4		Component of D-Axis Amortisseur Flux	p.u.
*A5		Component of Q-Axis Amortisseur Flux	p.u.
A7		Generator Saturation Correction Current	p.u.
A9	i_{ef}	Exciter Field Current	amps
E3	e_e	Exciter Output Voltage	volts
E4	e_{ed}	Exciter Output - A.C. Voltage	volts
E5	e_{ed}	Voltage Behind Exciter Transient Reactance	volts
E6	e_f	Generator Main Field Voltage	p.u.
*E7	i_o	Internal Parameter	
*SD	ψ_d	Armature Flux, D-Axis	p.u.
*SQ	ψ_q	Armature Flux, Q-Axis	p.u.
*SMC		Internal Parameter	
SM	ψ_{md}	Mutual Flux, D-Axis	p.u.
SN		Input to Saturation Table, FA1	p.u.
TD	T_D	Generator Output Torque	p.u.

*This output quantity is a state.

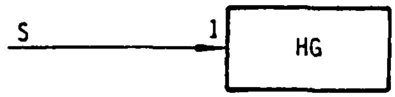
GENERATOR - EXCITER
INPUT

G8

PHYSICAL QUANTITY NAME	FIGURE 3.0-2 NAME	DESCRIPTION	UNITS
AFB	I_{fb}	Exciter Current Base Value	amps
EFB	E_{fb}	Exciter Voltage Base Value	volts
E1	e_d	Generator Terminal Voltage, D-Axis Component	p.u.
E2	e_q	Generator Terminal Voltage, Q-Axis Component	p.u.
E8	e_{ef}	Voltage Into Exciter Field	volts
FA1	f_1	Generator Saturation Function (Table)	
AM*		Degree of Interpolation for Table FA1	
FE4	f_{ef}	Exciter Saturation Function (Table)	
AN*		Degree of Interpolation for Table FE4	
K1	k_i	Exciter Current Rectification Constant	
K2	k_v	Exciter Voltage Rectification Constant	
PSM	PSM	1/Time Constant Default Value = 10000.0	rad/sec
R1	R_{kd}	Amortisseur Resistance, D-Axis Component	p.u.
R2	R_{kq}	Amortisseur Resistance, Q-Axis Component	p.u.
R3	R_f	Generator Field Resistance	p.u.
R4	R_a	Armature Resistance Per Phase	p.u.
R5	R_{ef}	Exciter Field Resistance	ohms
T1	τ_e	Exciter Field Open-Circuit Time Constant	secs
WO	ω_o	Base Frequency (WO = W_{zero})	rad/sec
W	ω	Input Speed	p.u.
X1	X_{f1}	Generator Field Leakage Reactance @ WO	p.u.
X2	X_{md}	Mutual Reactance, D-Axis @ WO	p.u.
X3	X_{mq}	Mutual Reactance, Q-Axis @ WO	p.u.
X4	X_{kd1}	Amortisseur Leakage Reactance, D-Axis @ WO	p.u.
X5	X_{kq1}	Amortisseur Leakage Reactance, Q-Axis @ WO	p.u.
X6	X_{a1}	Armature Leakage Reactance @ WO	p.u.
X7	X_{ed}	Synchronous Reactance, Exciter D-Axis	ohms
X8	X_{ed}	Transient Reactance, Exciter D-Axis	ohms
K3	K_3	Saturation Function Adjustment PSI-MD Default Value = 1.0	
K4	K_4	Saturation Function Adjustment ED Default Value = 0.3	

* A negative value prevents extrapolation beyond the table boundaries.

PROBABILITY DENSITY ANALYSIS **HG**



INPUT

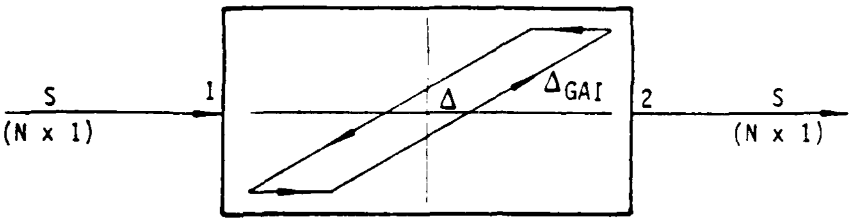
PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S	1	Input quantity to be monitored	
FUP		Upper limit for histogram	
FLO		Lower limit for histogram	
STR		Parameters to initialize calculation (DEFAULT provided)	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
F1 ~ F16		Output array containing histogram data	
FA		Measurement interval	

The input quantity is monitored during a SIMULATE analysis.
When time reaches TMAX, a histogram is produced with 16 intervals that span the range from FUP to FLO.
The histogram is drawn on page of the output history.

HYSTERESIS **HY**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input quantity	
GAI(N)		Gain	
DEL(N)		1/2 Histeresis Band Width	

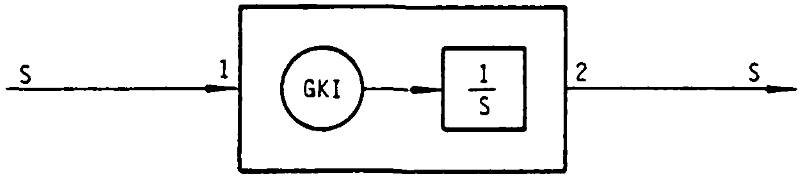
OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	Output quantity	
SL(N)		Previous input value	
CU(N)		Curve number	
TL		Previous time	
CPU		Precalculation indicator	

NOTE: N specifies the number of modes and is specified at Model Generation time.
The default value of N is 1.0.
This component is used in conjunction with ME, MM, MT, and MF.

INTEGRATOR

IN



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input	
GKI(N)		Integration gain	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
*S(N)	2	Output	

EQUATIONS:

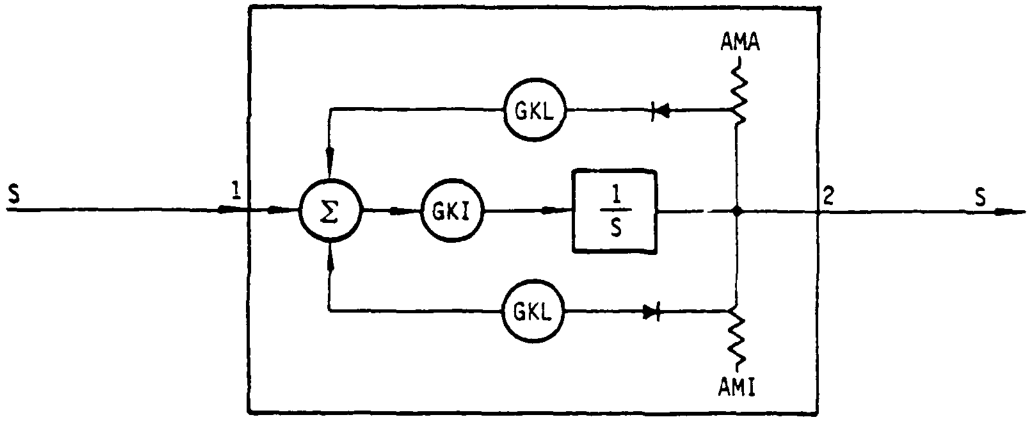
$$\dot{S}2 = GKI \cdot S1$$

*This output is a state

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

IT

INTEGRATOR WITH SATURATION



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input	
GKI(N)		Integration gain	
GKL(N)		Saturation limiter gain	
AMA(N)		Upper limit of output (Default = 10^{36})	
AMI(N)		Lower limit of output (Default = -10^{36})	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
*S(N)	2	Output	

EQUATIONS:

$\dot{S}2 = GKI[S1 - GKL(S2 - AMA)]$, if $S2 > AMA$

$\dot{S}2 = GKI \cdot S1$, if $AMI \leq S2 \leq AMA$

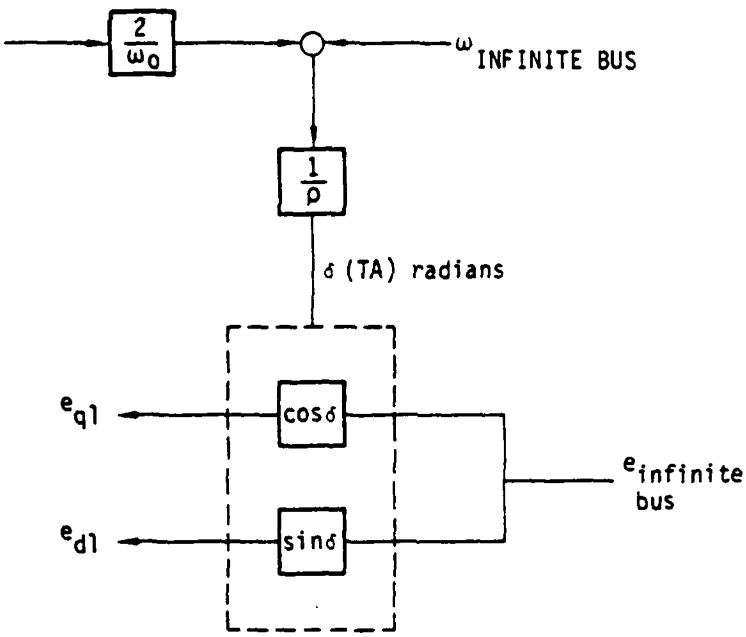
$\dot{S}2 = GKI[S1 - GKL(S2 - AMI)]$, if $S2 < AMI$

* This output is a state

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

11

INFINITE BUS



INPUT

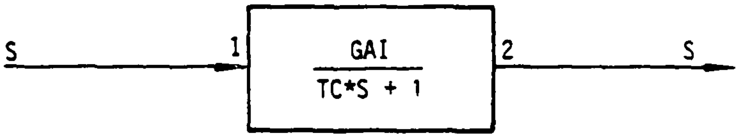
PHYSICAL QUANTITY NAME	FIGURE NAME	DESCRIPTION	UNITS
ER	E INFINITE BUS	VOLTAGE AT INFINITE BUS	PU
WI	ω INFINITE BUS	FREQUENCY AT INFINITE BUS	RAD/SEC
WO	ω_0	BASE FREQUENCY	PU
W1	ω_1	SHAFT ROTATION RATE	RAD/SEC

OUTPUT

PHYSICAL QUANTITY NAME	FIGURE NAME	DESCRIPTION	UNITS
E1	E_{d1}	DIRECT VOLTAGE	PU
E2	E_{q1}	QUADRATURE VOLTAGE	PU
WA		GENERATOR FREQUENCY	RAD/SEC
* TA	δ(TA)	TORQUE ANGLE	RADIANS
TX	-	TORQUE ANGLE	DEGREES

* THIS OUTPUT QUANTITY IS A STATE

FIRST ORDER LAG TRANSFER FUNCTION **LA**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input	
GAI(N)		Gain	
TC(N)		Time constant	seconds

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	Output	

EQUATIONS:

$$S2 = (GAI \cdot S1 - S2)/TC$$

NOTE: D.C. gain = GAI
and time constant = TC, seconds

infinite freq. gain = 0

pole location = $\frac{1}{TC}$ rad/sec

* This output is a state

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
YB, YBD		Side force coefficients:*	
		Beta and Beta dot coeff. (nondim.)	lb-sec/ft
		V and V dot coeff. (dim.)	lb-sec^2/ft
YP, YR		P and R angular rate coefficients	lb-sec/deg
YDR, YDA		Rudder and aileron coefficients	lb/deg
LB, LBD		Rolling moment coefficients:*	
		Beta and Beta dot coeff. (nondim.)	lb-sec
		V and V dot coeff. (dim.)	lb-sec^2
LP, LR		P and R angular rate coefficient	ft-lb-sec/deg
LDR, LDA		Rudder and aileron coefficients	ft-lb/deg
NB, NBD		Yawing moment coefficients:*	
		Beta and Beta dot coeff. (nondim.)	lb-sec
		V and V dot coeff. (dim.)	lb-sec^2
NP, NR		P and R angular rate coefficients	ft-lb-sec/deg
NDR, NDA		Rudder and aileron coefficients	ft-lb/deg
RUD, AIL		Control Surfaces:*	
		Rudder and aileron deflections	deg
UD, WD		Longitudinal accelerations:*	
		X and Z body axis acceleration	ft/sec^2
F(3)		External forces*	lbs
T(3)		External torques*	ft-lbs
UO(3)		X, Y, Z body axis velocities	ft/sec
WO(3)		X, Y, Z body axis angular rates	deg/sec
BE		Sideslip angle	deg
EV		Y body axis acceleration term for VD	ft/sec^2
VT		True airspeed	ft/sec
QS		Dynamic pressure x reference area	lbs
RW		Y body axis angular rate gust	deg/sec

VECTOR DEFINITIONS:

$$F(3) = \begin{pmatrix} FX \\ FY \\ FZ \end{pmatrix} \quad T(3) = \begin{pmatrix} TX \\ TY \\ TZ \end{pmatrix} \quad UO(3) = \begin{pmatrix} UO \\ VO \\ WO \end{pmatrix} \quad WO(3) = \begin{pmatrix} PO \\ QO \\ RO \end{pmatrix}$$

* Small Beta angle approximation

LD

INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
ID		Indicator function for coefficients 0 = body axis, dim. 1 = body axis, nondim. 2 = stability axis, dim. 3 = stability axis, nondim.	
CAL, SAL		Direction cosines for body or stability axes, depending on ID	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
FY	2	Y body axis force sum	lbs
VD		Y body axis acceleration	ft/sec^2
TX, TZ	2	X and Z axis (ROLL and YAW) moments	ft-lb

CONSTANTS

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
MA		Rigid body mass	slugs
B		Wing span	ft
XP*		X axis c.p. - c.g.	ft

LATERAL-AERODYNAMIC FORCES AND MOMENTS (Implicit form) **LD**

DIMENSIONAL EQUATIONS:

$$FY_{aero} = YB \cdot VO + YBD \cdot (\dot{V}+\dot{VW}) + YP \cdot PO + YR \cdot RO + YDR \cdot RUD + YDA \cdot AIL,$$

where

$$\dot{V} = VD = FY2/MA + EW$$

$$\dot{VW} = RW \cdot VT \cdot \pi/180$$

$$TX_{aero} = LB \cdot VO + LBD(\dot{V}+\dot{VW}) + LP \cdot PO + LR \cdot RO + LDR \cdot RUD + LDA \cdot AIL$$

$$TZ_{aero} = NB \cdot VO + NBD(\dot{V}+\dot{VW}) + NP \cdot PO + NR \cdot RO + NDR \cdot RUD + NDA \cdot AIL$$

NONDIMENSIONAL EQUATIONS:

$$FY_{aero} = QS \cdot (YB \cdot \hat{BE} + (YBD \cdot \dot{BETA} + YP \cdot \hat{P} + YR \cdot \hat{R})\ B/(2 \cdot VT) + YDR \cdot \hat{RUD} + YDA \cdot \hat{AIL}),$$

where

$$\dot{BETA} = \dot{V} \cdot (1 - \hat{BE}^2)/VT - \hat{BE}\ (UO \cdot UD + WO \cdot WD)/VT^2 + \hat{RW}^*$$

$$\hat{BE} = BE \cdot \pi/180, \text{ etc. for } \hat{P},\ \hat{R},\ \hat{RUD},\ \hat{AIL},\ \hat{RW}$$

$$TX_{aero} = QS \cdot B \cdot (LB \cdot \hat{BE} + (LBD \cdot \dot{BETA} + LP \cdot \hat{P} + LR \cdot \hat{R}) \cdot B/(2 \cdot VT) + LDR \cdot \hat{RUD} + LDA \cdot \hat{AIL})$$

$$TZ_{aero} = QS \cdot B \cdot (NB \cdot \hat{BE} + (NBD \cdot \dot{BETA} + NP \cdot \hat{P} + NR \cdot \hat{R}) \cdot B/(2 \cdot VT) + NDR \cdot \hat{RUD} + NDA \cdot \hat{AIL})$$

FORCE AND TORQUE SUM:

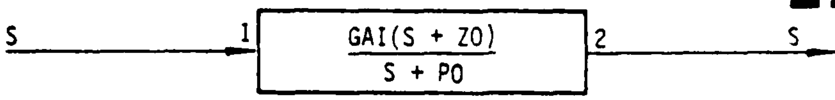
$$FY2 = FYaero + FY1$$

$$TX2 = \begin{cases} TX_{aero} + TX1 & ID = 0,\ 1 \\ TX_{aero} \cdot CAL - TZ_{aero} \cdot SAL + TX1 & ID = 2,\ 3 \end{cases}$$

$$TZ2 = \begin{cases} TZ_{aero} + TZ1 + XP \cdot FY_{aero} & ID = 0,\ 1 \\ TZ_{aero} \cdot CAL + TX_{aero} \cdot SAL + XP \cdot FY_{aero} & ID = 2,\ 3 \end{cases}$$

*Small Beta angle approximation

FIRST ORDER LEAD-LAG FUNCTION **LE**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input quantity	
GAI(N)		Infinite frequency gain	
Z0(N)		Numerator coefficient	rad/sec
P0(N)		Denominator coefficient	rad/sec

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
$\dot{X}$(N)		Intermediate quantity	
S(N)	2	Output quantity - variable	

EQUATIONS:

$$\dot{S}2 = GAI \cdot S1 + X1$$

$$\dot{X}1 = GAI \cdot S1 \cdot Z0 - S2 \cdot P0$$

NOTE:

d.c. gain = $\frac{GAI \cdot Z0}{P0}$

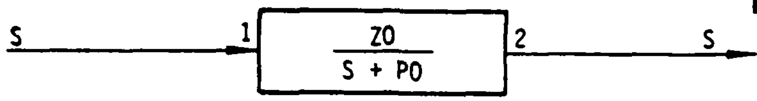
zero location = -Z0

infinite frequency gain = GAI

pole location = -P0

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

FIRST ORDER LAG TRANSFER FUNCTION **LG**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input quantity	
ZO(N)		Numerator coefficient	rad/sec
PO(N)		Denominator coefficient	rad/sec

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	Output quantity (state)	

EQUATION:

$$\dot{S}2 = ZO \cdot S1 - PO \cdot S2$$

NOTE:

d.c. gain $= \frac{ZO}{PO}$

time constant $= \frac{1}{PO}$

infinite frequency gain = 0

pole location = -PO

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

LEAD-LAG TRANSFER FUNCTION

LL

$$S \xrightarrow{\quad} {}_1\boxed{\dfrac{GAI(TC1 \cdot S + 1)}{TC2 \cdot S + 1}}{}_2 \xrightarrow{\quad} S$$

INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input quantity	
TC1(N)		Numerator time constant	sec
TC2(N)		Denominator time constant	sec
GAI(N)		Gain	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
*X1(N)		Intermediate quantity (state)	
S(N)	2	Output quantity (variable)	

EQUATIONS:

$$S2 = (X1 + S1 \cdot TC1 \cdot GAI)/TC2$$

$$\dot{X1} = GAI \cdot S1 - S2$$

NOTE:

d.c. gain = GAI

infinite gain = $\dfrac{GAI \cdot TC1}{TC2}$

zero location = $-\dfrac{TC1}{1}$, rad/sec

pole location = $-\dfrac{TC2}{1}$, rad/sec

*This output quantity is a state

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

INPUT LO

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
		X axis force coefficients:*	
XO		Bias coeff. for trim	lbs
XA		Alpha coeff. (nondim.)	
		Z axis velocity coeff. (dim.)	lb-sec/ft
XU		X axis velocity coeff.	lb-sec/ft
XDE		Elevator coefficent	lb/deg
		Z axis force coefficients:*	
ZO		Bias coeff. for trim	lbs
ZA, ZAD		Alpha and Alpha dot coeff. (nondim.)	
		Z axis velocity	lb-sec/ft
		and accel. coeff. (dim.)	lb-sec^2/ft
XQ		Z angular rate coeff.	lb-sec/deg
ZU		X axis velocity coeff.	lb-sec/ft
ZDE		Elevator coeff.	lb/deg
		Pitching moment coefficients:*	
MO		Bias coeff. for trim	ft-lb
MAL, MAD		Alpha and Alpha dot coeff. (nondim.)	
		Z axis velocity	lb-sec
		and accel. coeff. (dim.)	lb-sec^2
MQ		Q angular rate coeff.	ft-lb-sec/deg
MU		X axis velocity coeff.	lb-sec
MDE		Elevator coeff.	ft-lb/deg
		Constants:	
MA	1	Rigid body mass	slugs
C		Mean and aerodynamic chord	ft
XP*	1	X axis distance: c.p. - c.g.	ft
ID		Indicator function for coefficients	
		0 = body axis, dim.	
		1 = body axis, nondim.	
		2 = stability axis, dim.	
		3 = stability axis, nondim.	

*Default values = 0

INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
CAL, SAL		Direction cosines for stability, body axes	
		External forces and moments:*	
F(3)		X, Y, Z body axis forces	lbs
T(3)		Body axis (pitching) moment	ft-lb
		Aero-Variables:	
ELE*		Elevator deflection	deg
AL, ALP		Alpha in body and stability axes	deg
UO		X body axis velocity	ft/sec
UP, WP		X and Z perturbation velocities (nondim.)	
		X and Z stability axes velocities (dim.)	ft/sec
VT		True airspeed	ft/sec
QS		Dynamic pressure x reference area	lbs
QO, QW		Y body axis angular rate, rate gust	deg/sec
EU(3)		X, Y, Z axis accel. terms for UD, WD	ft/sec^2

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
FX, FZ	2	X and Z body axis force sum	lbs
TY	2	Y body axis (pitching) moment	ft-lbs
UD, WD		X and Z body axis acceleration	ft/sec^2
MA	2	Rigid body mass	slugs
XP	2	X axis distance: c.p. - c.g.	ft

*Default value = 0

VECTOR DEFINITIONS:

$$F(3) = \begin{pmatrix} FX \\ FY \\ FZ \end{pmatrix} \quad T(3) = \begin{pmatrix} TX \\ TY \\ TZ \end{pmatrix} \quad EU(3) = \begin{pmatrix} EU \\ EV \\ EW \end{pmatrix}$$

LONGITUDINAL AERO - FORCES AND MOMENTS
(Implicit Form)

LO

DIMENSIONAL EQUATIONS

$$FX_{aero} = XO + XA \cdot WP + XU \cdot UP + XDE \cdot ELE$$

$$FZ_{aero} = ZO + ZA \cdot WP + ZAD \cdot (\dot{W} + \dot{W}W) + ZQ \cdot QO + ZU \cdot UP + ZDE \cdot ELE,$$

where

$$\dot{W} = WD - UD \cdot SAL^{*}$$

$$\dot{W}W = -QW \cdot VT$$

$$TY_{aero} = MO + MAL \cdot WP + MAD \cdot (\dot{W} + \dot{W}W) + MQ \cdot QO + MU \cdot UP + MDE \cdot ELE$$

NONDEMENSIONAL EQUATIONS

$$FX_{aero} = QS \cdot (XO + XA \cdot \widehat{ALP} + XU \cdot UP + XDE \cdot \widehat{ELE})$$

$$FZ_{aero} = QS \cdot (ZO + ZA \cdot \widehat{ALP} + (ZAD \cdot (ALPHA - \widehat{QW}) + ZQ \cdot \widehat{QO}) \cdot C/(2 \cdot VT + ZU \cdot UP + ZDE \cdot \widehat{ELE}),$$

where

$$ALPHA = (WD - \widehat{AL} \cdot UD)/UO^{*}$$

$$\widehat{ALP} = ALP \cdot \pi/180, \text{ etc., for } \widehat{ELE}, \widehat{QW}, \widehat{QO}, \widehat{AL}$$

$$TY_{aero} = QS \cdot C \cdot (MO + MAL \cdot \widehat{ALP} + (MAD \cdot (ALPH - \widehat{QW}) + MQ \cdot \widehat{QO}) \cdot C/(2\ VT) + MU \cdot UP + MDE \cdot \widehat{ELE})$$

FORCE AND TORQUE SUM

$$FX_{sum} = FX_{aero} + FX1 \cdot CAL + FZ1 \cdot SAL$$

$$FZ_{sum} = FZ_{aero} + FZ1 \cdot CAL - FX1 \cdot SAL$$

$$FX2 = FX_{sum}\ CAL - FZ_{sum} \cdot SAL$$

$$FZ2 = FZ_{sum}\ CAL + FX_{sum} \cdot SAL$$

$$TY2 = TY_{aero} + TY1 - XP \cdot (FZ_{aero} \cdot CAL + FX_{aero} \cdot SAL)$$

ACCELERATIONS

$$UD = FX2/MA + EU$$

$$WD = FZ2/MA + EW$$

*Small alpha angle approximation.

LOAD

L2

INPUT

PHYSICAL QUANTITY NAME	FIGURE 3.0-3 NAME	DESCRIPTION	UNITS
ADI	i_{d1}	Load Current, D-Axis Component	p.u.
AQI	i_{q1}	Load Current, Q-Axis component	p.u.
RS2	R_{s2}	Linear Load, Simulated Open-Circuit Resistance	p.u.
RL	R_L	Linear Load Resistance	p.u.
RNL	R_{NL}	No-Load Shunt Resistance Default Value = 50.0	p.u.
XL	X_L	Linear Load Reactance	p.u.
XC	X_c	No-Load Shunt Capacitive Reactance Default Value = 50.0	p.u.
WO	ω_0	Base Frequency (WO = W_{zero})	rads/sec

OUTPUT

PHYSICAL QUANTITY NAME	FIGURE 3.0-3 NAME	DESCRIPTION	UNITS
*EDO	e_{d1}	Load Voltage, D-Axis Component	p.u.
*EQO	e_{q1}	Load Voltage, Q-Axis Component	p.u.
*ADS	i_{ds}	Intermediate Quantity (State)	p.u.
*AQS	i_{qs}	Intermediate Quantity (State)	p.u.
RTL		Intermediate Quantity	p.u.

EQUATIONS:

EDO = WO•XC•(-EDO/RNL+ADI-ADS+EQO/XC)

EQO = WO•XC•(-EQO/RNL+AQI-AQS-EDO/XC)

ADS = -ADS•WO•RTL/XL+EDO•WO/XL+AQS•WO

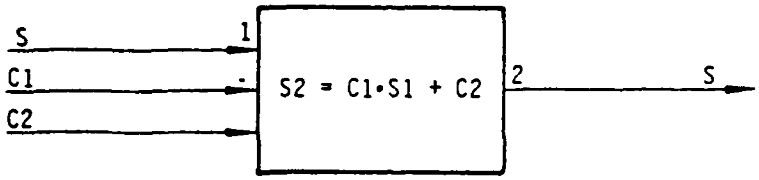
AQS = -AQS•WO•RTL/XL+EQO•WO/XL-ADS•WO

RTL = RS2+RL

*This output quantity is a state.

MA

MULTIPLY AND ADD



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input quantity	
C1(N)		Input quantity	
C2(N)		Input quantity	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	Output quantity	

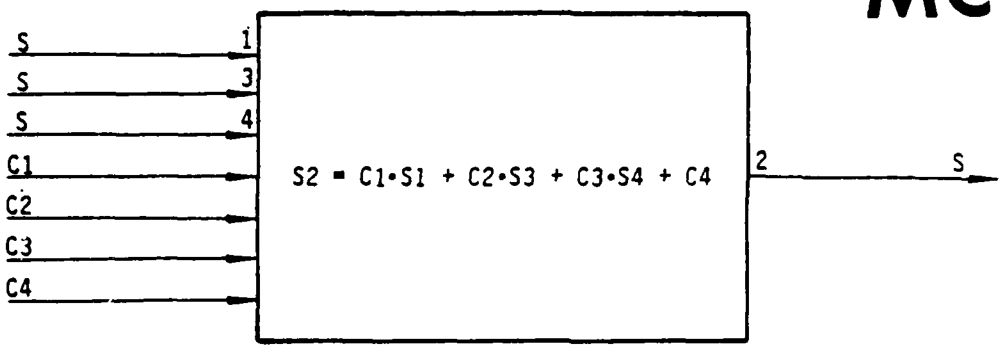
EQUATION:

$$S2 = C1 \cdot S1 + C2$$

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1

MULTIPLY AND ADD

MC



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input quantity	
S(N)	3	Input quantity	
S(N)	4	Input quantity	
C1(N)		Input quantity	
C2(N)		Input quantity	
C3(N)		Input quantity	
C4(N)		Input quantity	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	Output quantity	

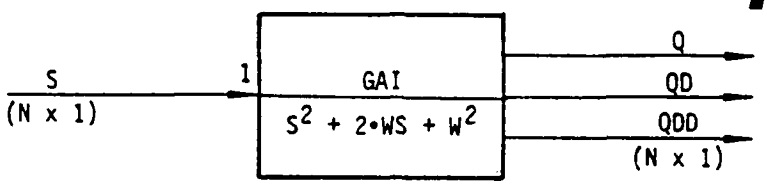
EQUATION:

$$S2 = C1 \cdot S1 + C2 \cdot S3 + C3 \cdot S4 + C4$$

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

STRUCTURAL MODE DYNAMICS

MD



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Mode excitation	
DMP(N)		Mode damping	
WN(N)		Mode natural frequency - $\overset{\omega}{W}$	rad/sec
GAI(N)		Mode gain at scale factor	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
Q(N)		Mode position (state)	
QD(N)		Mode velocity (state)	1/sec
QDD(N)		Mode acceleration	$1/sec^2$

EQUATIONS:

$$QDD(I) = GAI \cdot S(I) - WN(I) \cdot (WN(I) \cdot Q(I) + 2 \cdot DMP(I) \cdot QD(I))$$

$$\dot{Q}D(I) = QDD(I)$$

$$\dot{Q}(I) = QD(I) \qquad I = 1, 2, \ldots, N$$

Freezing Q(I) causes QD(I) to be frozen and QDD(I) to be set to zero, thus removing all effects of that mode from the model.

N specifies the number of modes, and is specified at Model Generation time. The default value of N is 1.0. This component is used in conjunction with ME, MM, MT, and MF.

ME

STRUCTURAL MODE EXCITATION

INPUTS

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
F(3)		Disturbance Force or Torque	lbs or ftlbs
DCM(N, 3)		Disturbance coefficient matrix	

OUTPUTS

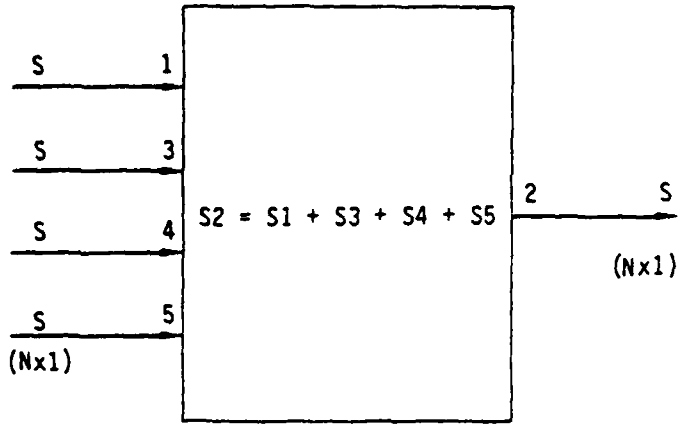
PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	Mode excitation	–

N specifies the number of modes and is specified at Model Generation time. The default value of N is 1.

This component is used in conjunction with MD, MM, MT, and MF.

MF

FOUR VECTOR SUM



INPUTS

PHYSICAL QUANTITY NAME	PORT	DESCRIPTION	UNITS
S(N)	1	Input quantity	–
S(N)	3	Input quantity	–
S(N)	4	Input quantity	–
S(N)	5	Input quantity	–

OUTPUTS

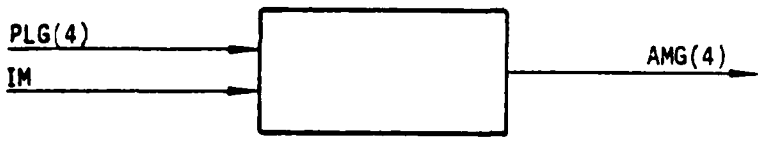
PHYSICAL QUANTITY NAME	PORT	DESCRIPTION	UNITS
S(N)	2	Output quantity	–

N specifies the number of modes and is specified at Model Generation time. The default of N is 1.

This component is used in conjunction with MD, MM, MT—MF.

EARTH'S MAGNETIC FIELD MODEL

MG



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
PLG(4)		Position in local geographic coordinates	
IM		Flag { = 0 Magnetic field in TESLA; = 1 Magnetic field in gauss	

OUTPUT

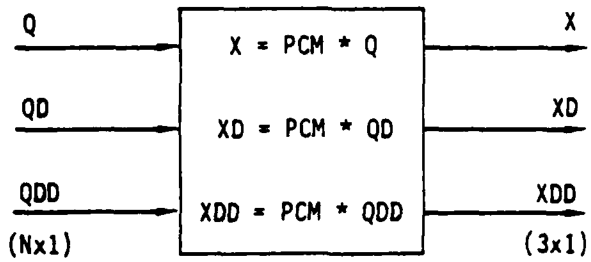
PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
AMG(4)		Magnetic field data	

VECTOR DESCRIPTION:

PLG(1) = Distance from geocenter, earth radii - dimensionless
PLG(2) = Co-latitude = $\pi/2$ - Geographic north latitude, radians
PLG(3) = Geographic east longitude, radians
PLG(4) = i , orbit inclination measured at ascending node, radians
AMG(1) = Magnitude of magnetic field, tesla or gauss
AMG(2) = Magnetic field along line of flight, tesla or gauss
AMG(3) = Magnetic field perpendicular to orbit plane, tesla or gauss
AMG(4) = Magnetic field along local vertical, tesla or gauss

MM

STRUCTURAL MODE MOTION



INPUTS

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
Q(N)		Mode position	–
QD(N)		Mode velocity	1/sec
QDD(N)		Mode acceleration	$1/sec^2$
PCM(3,N)		Position coefficient matrix	u *

OUTPUTS

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
X(3)		Position due to modes	u *
XD(3)		Velocity due to modes	u */sec
XDD(3)		Acceleration due to modes	$u\ */sec^2$

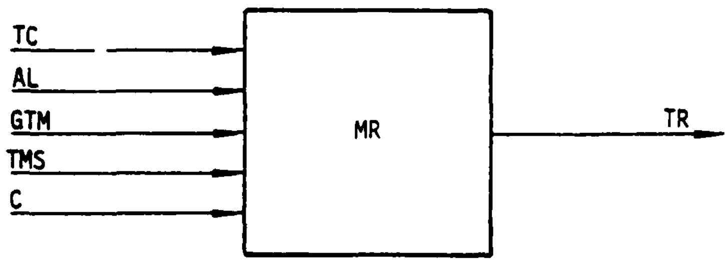
*Units depend on units used in PCM.

N specifies the number of modes and is specified at Model Generation time. The default value of N is 1.

This component is used in conjunction with ME, MD, MT, and MF.

MR

MOTOR RIPPLE



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
TC		Torque motor command	ma
AL		Torque motor angle	rad
GTM		Torque motor gain	ft-lb/ma
TMS		Torque motor saturation limit	ma
C		Array of Ripple Model coefficents and frequencies (See below)	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
TR		Motor torque	ft-lbs

EQUATIONS:

$$TCS = GTM \cdot SATUR\ (TC,\ TMS)$$

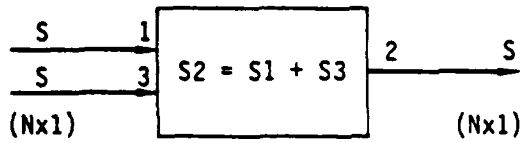
$$\begin{aligned} TR = {} & TCS \cdot (1. + C(4) \cdot SIN(C(14) \cdot AL) + C(5) \cdot COS(C(15) \cdot AL) \\ & + C(6) \cdot SIN(C(16) \cdot AL) + C(7) \cdot COS(C(17) \cdot AL) \\ & + (C(8) \cdot TCS \cdot TCS + C(9) \cdot ABS(TCS) + C(10)) \cdot SIN(C(18) \cdot AL) + C(11) \\ & + SIN(C(21) \cdot AL) + C(12) \cdot COS(C(22) \cdot AL) + C(13) \cdot SIN(C(23) \cdot AL)) \end{aligned}$$

RIPPLE MODEL COEFFICIENTS & FREQUENCIES: C SUBSCRIPT USAGE:

RIPPLE MODEL COMPONENT	COEFFICIENT	FREQUENCY
Hall probe null	4	14
Common node	5	15
Hall probe placement	6	16
Unequal gains	7	17
Magnetic field	8, 9, 10	18
Offset currents	11, 12	21, 22
Reluctance (Cogging)	13	23

MT

TWO VECTOR SUM



INPUTS

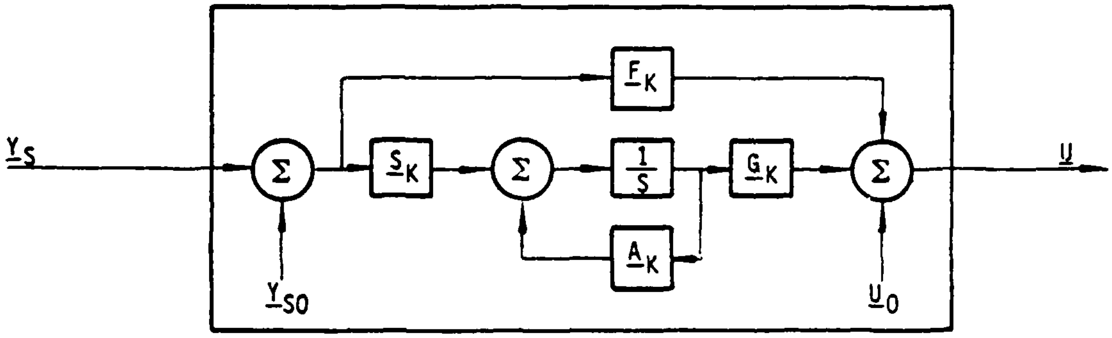
PHYSICAL QUANTITY NAME	PORT	DESCRIPTION	UNITS
S(N)	1	Input quantity	–
S(N)	3	Input quantity	–

OUTPUTS

PHYSICAL QUANTITY NAME	PORT	DESCRIPTION	UNITS
S(N)	2	Output quantity	–

N specifies the number of modes and is specified at Model Generation time. The default value of N is 1. This component is used in conjunction with MT ← ME

OPTIMAL CONTROLLER **OC**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
		All optimal controller inputs are defined via the O.C. INPUTS command in the EASY Model Generation Program.	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
		All optimal controller outputs are defined via the O.C. OUTPUTS command in the EASY Model Generation Program.	

NOTE: Due to its very general nature, the O.C. component is specified by a special set of Model Generation and Analysis commands, which all start with the letters O.C. (See Section 4.13)

POWER FACTOR CONTROLLER

INPUT

PHYSICAL QUANTITY NAME	FIGURE NAME	DESCRIPTION	UNITS
ED	ED	D AXIS VOLTAGE	PER UNIT
EQ	EQ	Q AXIS VOLTAGE	PER UNIT
AD	AD	D AXIS CURRENT	PER UNIT
AQ	AQ	Q AXIS CURRENT	PER UNIT
X1	X1	LEAD TIME CONSTANT	SEC
X2	X2	LEAD TIME CONSTANT	SEC
X3	X3	INTEGRAL GAIN (INVERSE)	-
X4	X4	LAG TIME CONSTANT	SEC
PFR	PFR	POWER REFERENCE FACTOR	
AB	AB	BASE LINE CURRENT	AMPS
VB	VB	BASE LINE VOLTAGE	(SEE CODE)
CMA	CMA	OUTPUT LIMITER (MAX)	PER UNIT
CMI	CMI	OUTPUT LIMITER (MIN)	PER UNIT
G1	G1	SATURATION SLOPE	-

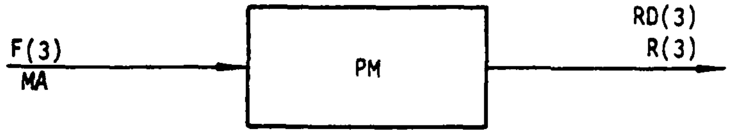
OUTPUT

PHYSICAL QUANTITY NAME	FIGURE NAME	DESCRIPTION	UNITS
* B1	-	INTERMEDIATE STATE	
* B2	-	INTERMEDIATE STATE	
ARO	ARO	REAL CURRENT	AMPS
AI1	AI1	REACTIVE CURRENT	AMPS
AT	AT	TOTAL CURRENT	AMPS
VPF	VPF	OUTPUT TO VOLTAGE REGULATOR	(SEE CODE)
PFL	PFL	LINE POWER FACTOR	-
FIN	FIN	ERROR INPUT	PER UNIT
FO	FO	LEAD LAG OUTPUT	PER UNIT

* THESE OUTPUT QUANTITIES ARE STATES

POINT MASS IN GRAVITY FIELD

PM



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
F(3)		External Force Vector, inertial axes	lbs
MA		Mass	slugs
*LA		Initial latitude	deg
*LO		Initial longitude	deg
ALT		Initial altitude	feet
**TI		Initial time	hour
***DA		Initial date - Julian day	day
VEL		Initial velocity	ft/sec
*AZI		Initial horizontal flight path angle (azimuth)	deg
*GAM		Initial vertical flight path angle	deg

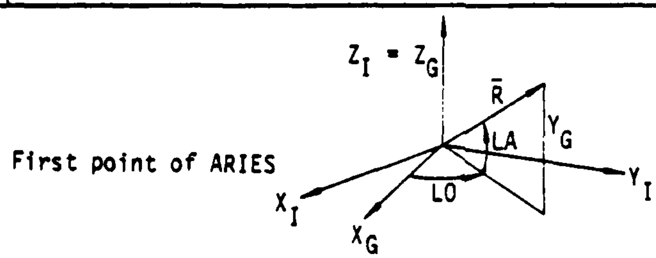
* Default values of zero are provided for these quantities

** Default value of 12 is provided for TI

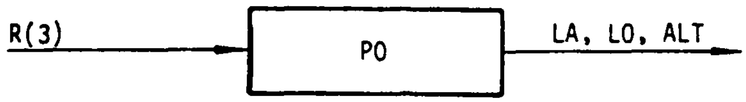
*** Default value of 80 is provided for DA

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
R(3)		Position vector, inertial axes	ft
RD(3)		Velocity vector, inertial axes	ft/sec



POSITION AND ORIENTATION OF POINT MASS **PO**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
R(3)		Position vector, inertial axes	ft
RD(3)		Velocity vector, inertial axes	ft/sec
A(3,3)		Inertial to Body Axis Transformation Matrix	
TI		Initial time	hours
DA		Initial date - Julian days	days

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
LA		Latitude	day
LO		Longitude	day
ALT		Altitude	ft
AZI		Azimuth angle, 0 = North & clockwise	deg
GAM		Vertical flight path angle, + = pitch up	deg
EA(3)		Euler angles - Local Horizontal to Body Axes	

EQUATIONS:

$$LO = \tan^{-1}\left(\frac{R(2)}{R(1)}\right) + \frac{360}{365}\left(80 - DA\right) - \frac{360}{12}\left(TI - 12 - \frac{TIME}{3600}\right)$$

$$ALT = |R| - 20927491.$$

$$LA = \sin^{-1}\left(\frac{R(3)}{|R|}\right)$$

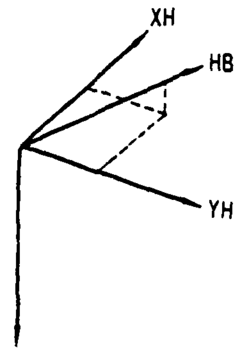
PO

Vehicle Attitude relative to Local Horizontal Transformation from Initial to Body Axes is given:

$$T_{BI} = T_1\,(\phi)\;T_2\,(\theta)\;T_3\,(\psi)\;T_2\,(-90 - \Phi)\;T_3\,(\alpha - \lambda + \Lambda)$$

Separate Transformation from Local Horizontal to Body Axes

$$D_{BH} = T_1\,(\phi)\;T_2\,(\theta)\;T_3\,(\psi) = T_{BI}\;T_3\,(\lambda - \alpha - \Lambda)\;T_2\,(90 + \Phi)$$



$$\theta = \sin^{-1}\left(-d_{13}\right)$$

$$\phi = \tan^{-1}\left(\frac{d_{23}}{d_{33}}\right)$$

$$\psi = \tan^{-1}\left(\frac{d_{12}}{d_{11}}\right)$$

PO

Calculation of Flight Path Angle and Euler Angles relating Body Axes to Local Horizontal Axes:

Flight Path Angles

Given: $\dot{R}_I$ - velocity vector inertial coordinates

Φ - latitude

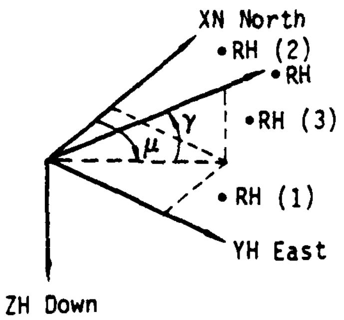
Λ - longitude

α - time angle

λ - date angle

Transform velocity vector into Local Horizontal Axes

$$R_H = T_2\ (-90 - \Phi)\ T_3\ (\alpha - \lambda + \Lambda)\ \dot{R}_I$$



μ = azimuth - horizontal flight path angle

γ = vertical flight path angle

$$AZI = \mu = \tan^{-1}\left(\frac{\dot{R}_H\ (2)}{\dot{R}_H\ (1)}\right)$$

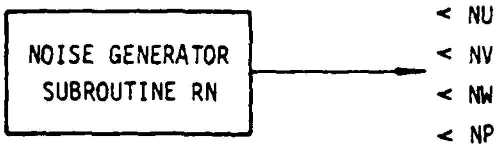
$$GAM = \gamma = \tan^{-1}\left(\frac{-\dot{R}_H\ (3)}{\left(\dot{R}_H^2\ (1) + \dot{R}_H^2\ (2)\right)^{\frac{1}{2}}}\right)$$

AD-A096 597 BOEING MILITARY AIRPLANE CO SEATTLE WA F/G 1/3
ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM. VOLUME --ETC(U)
SEP 80 C L WEST, B R UMMEL, R F YURCZYK F33615-79-C-3407
UNCLASSIFIED AFWAL-TR-80-3014-VOL-1 NL

7 OF 8

AD A
096597

NOISE GENERATOR FOR WIND MODEL **RA**



OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
NU, NV, NW		Noise samples for U, V, W gust velocities	
NP		Noise sample for P angular rate gust	

METHOD:

Call RN(VAR, DUM, SIG, AMN)

where

VAR = Gaussian random output variable

DUM = Internal variable to start RN

SIG = Standard deviation of VAR = $\sqrt{2.0}/\Delta$; where

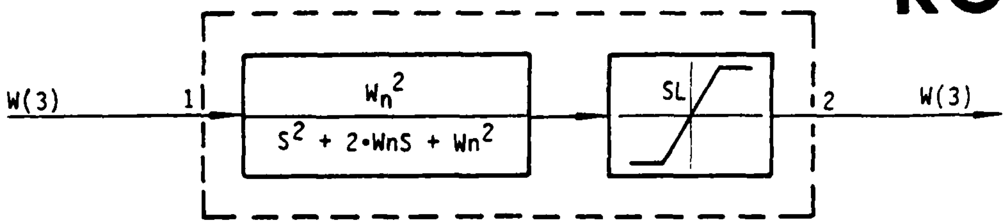
Δ = integrator stepsize

AMN = Var mean value = 0

NOTE: RA can only be used with the fixed step integrator which is specified by the command: INT MODE = 3 or 4

RATE GYRO PACKAGE

RG



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
W(3)	1	Three axis angular rates	rad/sec
SL		Rate gyro saturation level (Same for all axes)	rad/sec
DMP		Rate gyro damping coefficient, ζ	
WN		Rate gyro natural frequency, W_n	rad/sec

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
W(3)	2	Three axis angular rates as output by gyros-states	rad/sec
WX(3)		Intermediate states associated with each rate gyro	

EQUATIONS:

$FB = W2(I)$

$IF(|W2(I)|>SL), FB = 100 \cdot (W1(I) - SIGN(SL, W2(I)) + SIGN(SL, W2(I))$

$\dot{W}X(I) = (W1(I) - FB) \cdot WN$

$\dot{W}2(I) = (WX(I) - 2 \cdot DMP \cdot FB) \cdot WN \quad I = 1, 2, 3$

Saturation of output state is accomplished by increasing feedback gain by 100 if output exceeds saturation limit.

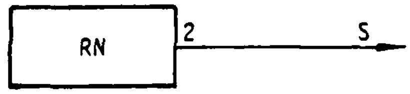
VECTOR DEFINITIONS:

$$W1(3) = \begin{pmatrix} P1 \\ Q1 \\ R1 \end{pmatrix} \quad W2(3) = \begin{pmatrix} P2 \\ Q2 \\ R2 \end{pmatrix} \quad WX(3) = \begin{pmatrix} PX \\ QX \\ RX \end{pmatrix}$$

NOTE: Component XP should be used to convert to and from body axes to gyro axes.

RANDOM NUMBER GENERATOR

RN



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
AX		Seed (Default = 43146971.)	
SIG		Requested standard deviation	
MN		Requested mean	

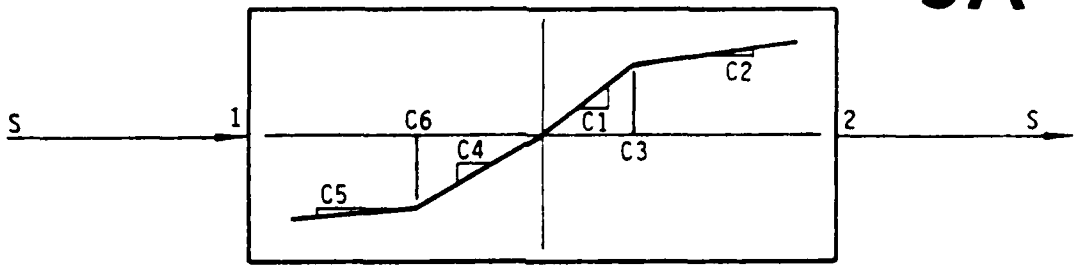
OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S	2	Random number output	

RN generates a normally distributed random number each time it is called.
The seed, AX, should be an odd number greater than one.
This component is automatically disabled for all analyses except SIMULATE.

SATURATION FUNCTION

SA



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S	1	Input quantity	
C1		Slope $0 < S1 < C3$	
C2		Slope $S1 > C3$	
C3		Positive saturation intercept	
C4		Slope $0 > S1 > C6$	
C5		Slope $S1 < C6$	
C6		Negative saturation intercept	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S	2	Output quantity	

EQUATIONS:

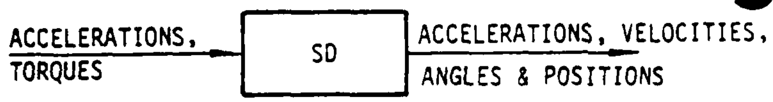
$$S2 = C1 \cdot C3 + C2 \cdot (S1 - C3) \quad \text{if } S1 > C3$$

$$S2 = C1 \cdot S1 \quad \text{if } 0 < S1 < C3$$

$$S2 = C4 \cdot S1 \quad \text{if } 0 > S1 > C6$$

$$S2 = C4 \cdot C6 + C5 \cdot (S1 - C6) \quad \text{if } S1 < C6$$

SIX-DEGREE-OF-FREEDOM RIGID BODY DYNAMICS **SD**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
UD, VD, WD		X, Y, Z body axis linear accelerations	ft/sec^2
TX, TY, TZ		X, Y, Z body axis torques	ft-lbs
IXX, IYY, IZZ		X, Y, Z body axis moments of inertia	$slug\text{-}ft^2$
IXZ, IXY, IYZ		Cross products of inertia	$slug\text{-}ft^2$

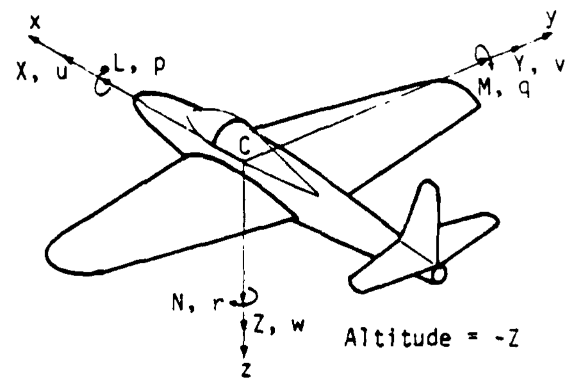
OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
U(3)		X, Y, Z body axis linear velocities	ft/sec
W(3)		X, Y, Z body axis angular rates	deg/sec
EA(3)		Euler angles, body to inertial axes	deg
XD, YD		Horizontal position rates	ft/sec
*ALT		Vertical altitude from sea-level	ft
WD(3)	2	X, Y, Z body axis angular accelerations	deg/sec^2

ASSUMPTIONS:

1. Constant gravity, flat-earth model.

*These output quantities are states.



SD

SIX DEGREE OF FREEDOM EQUATIONS OF MOTION

LINEAR VELOCITY EQUATIONS

$\dot{U}$ = UD(1) = UDOT(1)

$\dot{V}$ = UD(2) = UDOT(2)

$\dot{W}$ = UD(3) = UDOT(3)

ANGULAR VELOCITY EQUATIONS

```
TXE = TX+YZI*(Q1**2-R1**2)+XZI*P1*Q1-XYI*R1*P1
        +(YYI-ZZI)*Q1*R1

TYE = TY+XZI*(R1**2-P1**2)+XYI*Q1*R1-YZI*P1*Q1
        +(ZZI-XXI)*R1*P1
TZE = TZ+XYI*(P1**2-Q1**2)+YZI*R1*P1-XZI*Q1*R1
        +(XXI-YYI)*P1*Q1

DETI = XXI*(YYI*ZZI-YZI**2)-XYI*(YZI*XZI+ZZI*XYI
        -XZI*(XYI*YZI+YYI*XZI)

WD(1) = (TXE*(YYI*ZZI-YZI**2)+TYE*(XYI*ZZI
        +YZI*XZI)+TZE*(XYI*YZI+YYI*XZI)/DETI

WD(2) = (TXE*(XYI*ZZI+YZI*XZI)+TYE*(XXI*ZZI
        -XZI**2)+TZE*(XXI*YZI+XYI*XZI))/DETI

WD(3) = (TXE*(XYI*YZI+YYI*XZI)+TYE*(XXI*YZI
        +XYI*XZI)+TZE*(XXI*YYI-XYI**2))/DETI
```

ANGULAR POSITION EQUATIONS

```
PITD = EAD(2) = W(2)*CR-W(3)*SR

PSID = W(2)*SR+W(3)*CR)/CP
       EAD(3) = PSID

ROLD = EAD(1) = W(1) + PSID*SP
```

SD

LINEAR POSITION EQUATIONS

XD = CY(CP*U(1) +(-SY*CR + CY*SPSR)*U(2) + (SY*SR + CY*SPCR)*Y(3)

YD = SY*CP*U(1) + (CY*CR + SY*SPSR)*U(2) + (-CY*SR + SY*SPCR)*U(3)

ZD = SP*U(1)-CP*SR*U(2)-CP*CR*U(3)

The following abbreviations are used in these equations:

SR = SIN(ROL)	CR = COS(ROL)
SP = SIN(PIT)	CP = COS(PIT)
SY = SIN(YAW)	CY = COS(YAW)

SPSR = SP*SR
SPCR = SP*CR

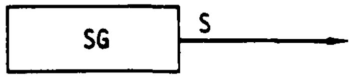
VECTOR DEFINITIONS:

$$UD(3) = \begin{pmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{pmatrix} \qquad U(3) = \begin{pmatrix} U \\ V \\ W \end{pmatrix}$$

$$WD(3) = \begin{pmatrix} \dot{P1} \\ \dot{Q1} \\ \dot{R1} \end{pmatrix} \qquad W(3) = \begin{pmatrix} P1 \\ Q1 \\ R1 \end{pmatrix} \qquad EA(3) = \begin{pmatrix} ROL \\ PIT \\ YAW \end{pmatrix}$$

SERVO ANALYZER SIGNAL GENERATOR

SG



(This component is used with component SS)

INPUTS

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
FR1		Initial, (lower), frequency	hertz
FR2		Final, (upper), frequency	hertz
AM1		Initial, (lower), amplitude	–
AM2		Final, (upper), amplitude	–

OUTPUTS

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S		Test Si Test signal	–
F		Test signal frequency	hertz
LGF		Log of test signal frequency	–
AMP		Amplitude of test signal	–

Equations:

$S = AMP \sin (2\ Ft)$

Frequency scan occurs if:

$FR2 > FR1$

Amplitude scan occurs if:

$FR2 \leq FR1$ or $FR2 = .99999$

WARNING

This component operates only with fixed step Huen integrator INTMODE = 3.
Only one SG Component can appear in a given model.

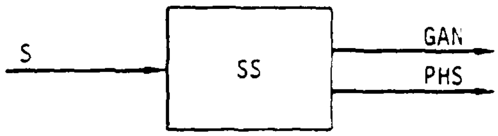
For frequency scans the following guidelines have been found useful in selecting simulation duration and step size.

$$TMAX \geq \frac{130}{FR1} * (\text{No. decades scanned})$$

$$TINC < \frac{1}{30 * FR2}$$

SS

SERVO ANALYZER



(This component is used with component SG)

INPUTS

PHYSICAL QUANTITY NAMES	PORT NO.	DESCRIPTION	UNITS
S		Test system output signal	

OUTPUTS

PHYSICAL QUANTITY NAMES	PORT NO.	DESCRIPTION	UNITS
GAN		Gain	db.
PHS		Phase	degrees
SI *		sine integrator	–
CI *		cosine integrator	–
SAV		sine (in phase) average value	–
CAV		cosine (quad phase) average value	–
CPU		signal used to initialize component	–

*These quantities are states

Equations:

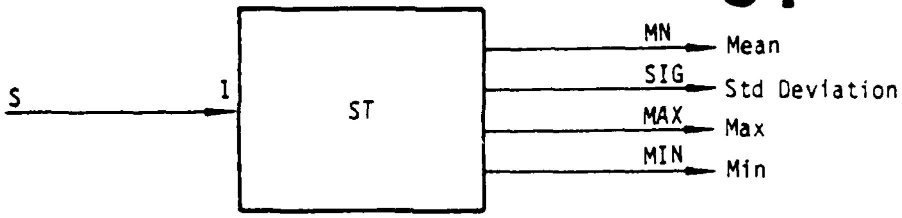
$$GAN = 20\log\left(\frac{2(SAV^2 + CAV^2)^{\frac{1}{2}}}{AMP^2}\right)$$

$$PHS = \tan^{-1}\left(\frac{CAV}{SAV}\right)$$

Several SS components can be used simultaneously with one S6 signal generator.

STATISTICAL ANALYSIS

ST



INPUT

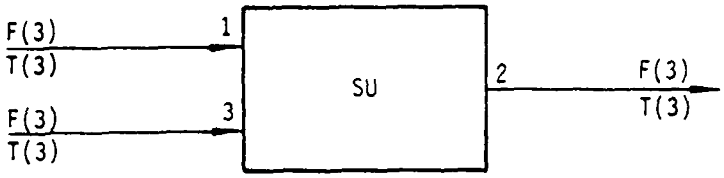
PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S	1	Input quantity to be monitored	
STR		Parameter to utilize calculations (Default provided)	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
MN		Running mean of input quantity	
MAX		Maximum value of input quantity	
MIN		Minimum value of input quantity	
SIG		Running standard deviation of input quantity - rms	

The measure of mean standard deviation, maximum, and minimum will start at the beginning of each SIMULATE analysis.

SUM TWO SETS OF 3-AXIS FORCES AND TORQUES **SU**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
F(3)	1	X, Y, Z body axis input forces, port 1	lbs
T(3)	1	X, Y, Z body axis input torques, port 1	ft-lbs
F(3)	3	X, Y, Z body axis input forces, port 3	lbs
T(3)	3	X, Y, Z body axis input torques, port 3	ft-lbs

OUTPUT

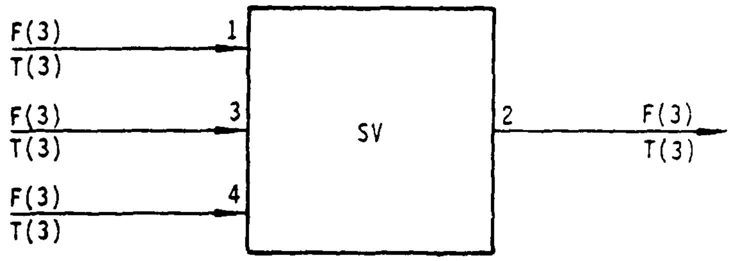
PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
F(3)	2	X, Y, Z body axis output forces, port 2	lbs
T(3)	2	X, Y, Z body axis output torques, port 2	ft-lbs

EQUATIONS:

F2(I) = F1(I) + F3(I)

T2(I) = T1(I) + T3(I)   I = 1, 2, 3

SUM THREE SETS OF 3-AXIS FORCES AND TORQUES **SV**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
F(3)	1	X, Y, Z body axis input forces, port 1	lbs
T(3)	1	X, Y, Z body axis input torques, port 1	ft-lbs
F(3)	3	X, Y, Z body axis input forces, port 3	lbs
T(3)	3	X, Y, Z body axis input torques, port 3	ft-lbs
F(3)	4	X, Y, Z body axis input forces, port 4	lbs
T(3)	4	X, Y, Z body axis input torques, port 4	ft-lbs

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
F(3)	2	X, Y, Z body axis output forces, port 2	lbs
T(3)	2	X, Y, Z body axis output torques, port 2	ft-lbs

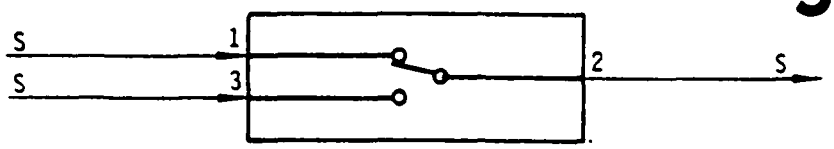
EQUATIONS:

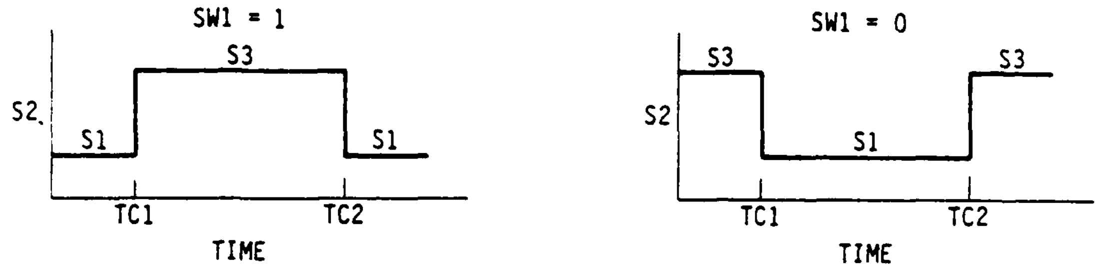
$$F2(I) = F1(I) + F3(I) + F4(I)$$

$$T2(I) = T1(I) + T3(I) + T4(I) \qquad I = 1, 2, 3$$

SW

SWITCH





The switching operation may be controlled by either time or the input parameter SW1. The time dependence may be eliminated by setting TC1 = 10^{36}

INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input to switch	
S(N)	3	Input to switch	
SW1		Switch control parameter	
TC1		Time for first switching	sec
TC2		Time for second switching	sec

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	Output from switch	

EQUATIONS:

$S2 = S1$ if $SW1 = 1$ and $t < TC1$ or $t > TC2$ or if $SW1 = 0$ and $TC1 < t < TC2$

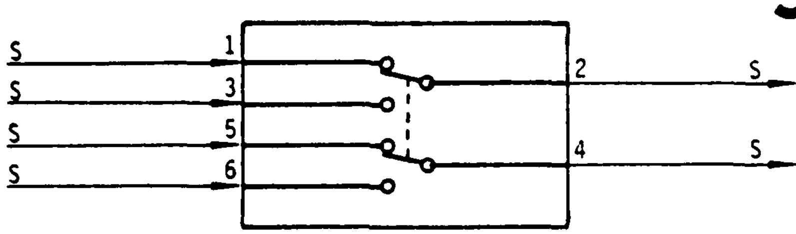
$S2 = S3$ if $SW1 = 0$ and $t < TC1$ or $t > TC2$ or if $SW1 = 1$ and $TC1 < t < TC2$

where; t = TIME, seconds

N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

TWO POLE SWITCH

SX



NOTE: See SW for switch control logic.

INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input to switch 1	
S(N)	3	Input to switch 1	
S(N)	5	Input to switch 2	
S(N)	6	Input to switch 2	
SW1		Switch control parameter	
TC1		Time for first switching	sec
TC2		Time for second switching	sec

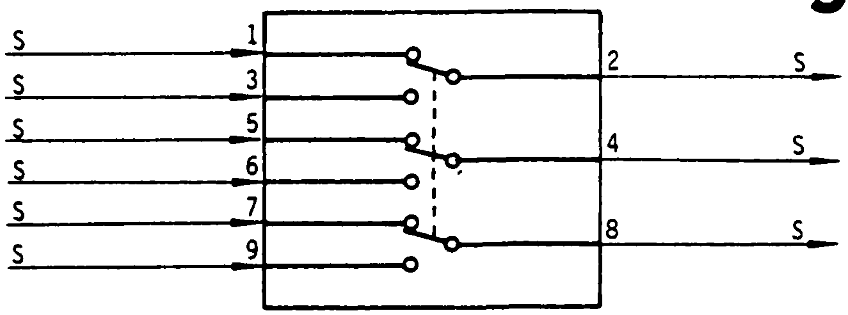
OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	Output from switch 1	
S(N)	4	Output from switch 2	

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

THREE POLE SWITCH

SY



NOTE: See SW for switch control logic.

INPUT

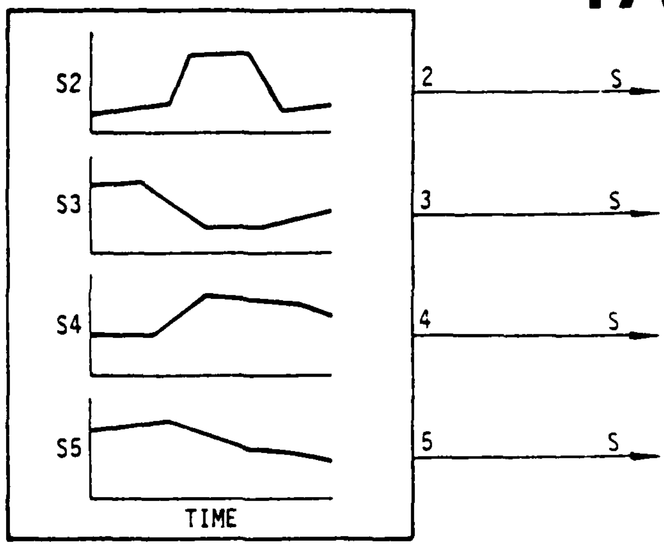
PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input to switch 1	
S(N)	3	Input to switch 1	
S(N)	5	Input to switch 2	
S(N)	6	Input to switch 2	
S(N)	7	Input to switch 3	
S(N)	9	Input to switch 3	
SW1		Switch control parameter	
TC1		Time for first switching	sec
TC2		Time for second switching	sec

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	Output from switch 1	
S(N)	4	Output from switch 2	
S(N)	8	Output from switch 3	

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

FOUR TABULAR FUNCTIONS OF TIME **TA**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
A2T		Tabular data describing S2 vs. time	
B2T		Tabular data describing S3 vs. time	
C2T		Tabular data describing S4 vs. time	
D2T		Tabular data describing S5 vs. time	

OUTPUT

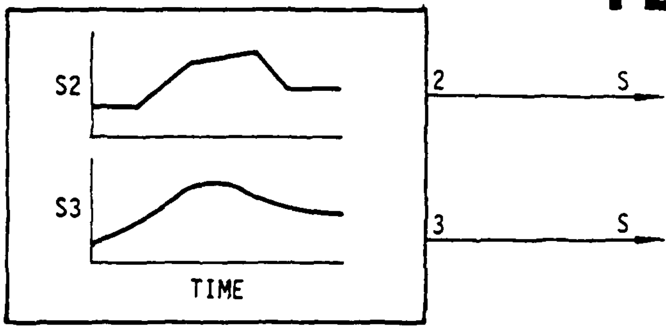
PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S	2	Output quantity	
S	3	Output quantity	
S	4	Output quantity	
S	5	Output quantity	

EQUATIONS:

S2 - A2T(t)
S3 = B2T(t)
S4 = C2T(t)
S5 = D2T(t)

NOTE: 15 points are allowed per table. Linear Interpolation is used between points. The last point in the table is used for values of time outside the table range

TWO TABULAR FUNCTIONS OF TIME **TB**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
A2T		Tabular data describing S2 vs. time	
P2T		Tabular data describing S3 vs. time	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S	2	Output quantity	
S	3	Output quantity	

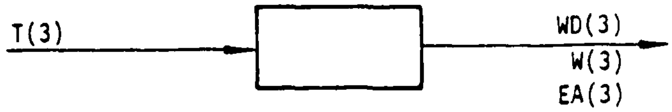
EQUATIONS:

$$S2 = A2T(t)$$

$$S3 = B2T(t)$$

NOTE: 15 points are allowed per table. Linear Interpolation is used between points. The last point in the table is used for values of time outside the table range.

THREE-DEGREE-OF-FREEDOM RIGID BODY **TD**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
T(3)		X, Y, Z body axis torques	ft-lb
IXX, IYY, IZZ		X, Y, Z body axis moments of inertia	slug-ft^2

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
W(3)		X, Y, Z body axis angular rates	rad/sec
EA(3)		Euler angles, body to fixed axes	rad
WD(3)		X, Y, Z body axis angular accelerations	rad/sec^2

ASSUMPTIONS:

1. Body axes are principal axes, i.e., products of inertia = 0
2. Body moments of inertia are constant
3. Euler angle sequence, body to fixed axes = roll, pitch, yaw.

TD

THREE DEGREE OF FREEDOM RIGID BODY

Angular Velocity Equations

$\dot{P}$ = PD = (TX - Q*R(ZZI - YYI))/XXI

$\dot{Q}$ = ZD = (TY - P*R*(XXI - ZZI))/YYI

$\dot{R}$ = RD = (TZ - Q*P*(YYI - XXI))/ZZI

Angular Position Equations

$\dot{PIT}$ = Q*COS(ROL) - R*SIN(ROL)

$\dot{YAW}$ = (Q*SIN(ROL) + R*COS(ROL))/COS(PIT)

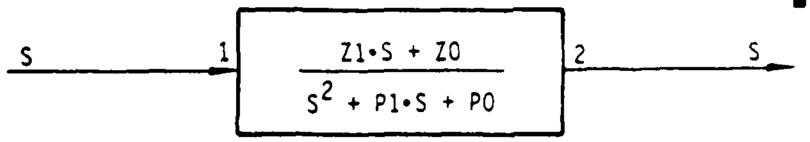
$\dot{ROL}$ = P + $\dot{YAW}$*SIN(PIT)

Vector Definitions:

$$T(3) = \begin{pmatrix} TX \\ TY \\ TZ \end{pmatrix} \qquad W(3) = \begin{pmatrix} P \\ Q \\ R \end{pmatrix} \qquad EA(3) = \begin{pmatrix} ROL \\ PIT \\ YAW \end{pmatrix} \qquad WD(3) = \begin{pmatrix} PD \\ QD \\ RD \end{pmatrix}$$

TRANSFER FUNCTION

TF



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	1	Input quantity	
Z0(N)		Numerator coefficient	
Z1(N)		Numerator coefficient	
P0(N)		Denominator coefficient	
P1(N)		Denominator coefficient	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
S(N)	2	Output quantity (state)	
X1(N)		Intermediate state (state)	

EQUATIONS:

$$\dot{X}1 = Z0 \cdot S1 - P0 \cdot S2$$

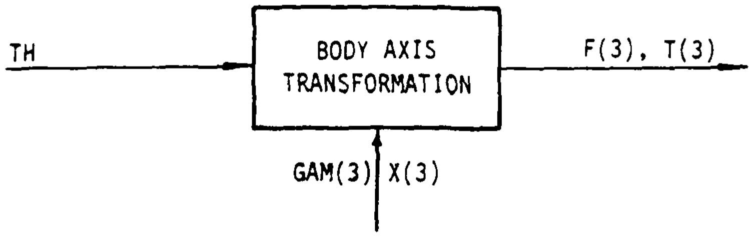
$$\dot{S}2 = X1 + Z1 \cdot S1 - P1 \cdot S2$$

NOTE: d.c. gain = $\frac{Z0}{P0}$

infinite frequency gain = 0

NOTE: N may be specified at Model Generation time to allow inputs and outputs to be N dimensional vectors. Default value of N is 1.0

ENGINE THRUST BODY AXIS TRANSFORM **TG**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
TH		Engine thrust	lbs
GAM(3)		X, Y, Z body axis direction cosines	
X(3)		X, Y, Z thrust location components	ft

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
F(3)		X, Y, Z body axis forces	lbs
T(3)		X, Y, Z body axis torques	ft-lbs

EQUATIONS:

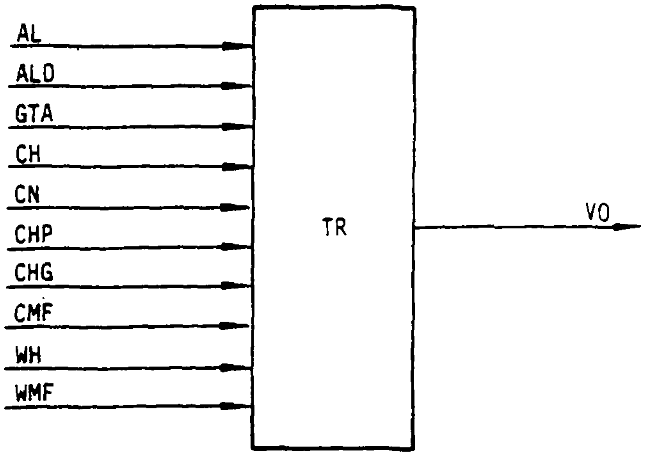
$$F(I) = TH \cdot GAM(I)$$

$$\overline{T} = \overline{X} \times \overline{F} \text{ (vector cross product)}$$

$$I = 1, 2, 3$$

TACHOMETER RIPPLE EFFECTS

TR



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
AL		Shaft angle	rad
ALD		Shaft rate	rad/sec
GTA		Tachometer gain	volt/rad/sec
CH		Hall probe null coefficient	
CN		Common node coefficient	
CHG		Unequal gain coefficient	
CMF		Magnetic field coefficient	
WH		Hall probe frequency	rad/sec
WMF		Magnetic field frequency	rad/sec

OUTPUT

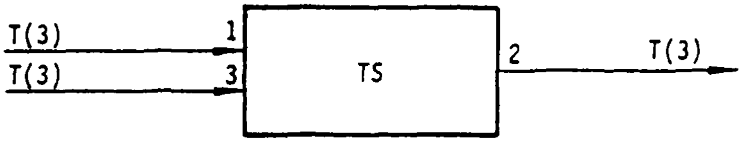
PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
VO		Tachometer output voltage	volt

EQUATIONS:

WHAL = WH•AL   WHPAL = WHP•AL

VO = GTA•ALD•(1. + CH•SIN(WHAL) + CN•COS(WHAL)
+ CHP•SIN(WHPAL + CHG•COS(WHPAL) + CMF•SIN(WMF•AL))

SUM TWO SETS OF 3-AXIS TORQUES **TS**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
T(3)	1	X, Y, Z body axis input torques, port 1	ft-lbs
T(3)	3	X, Y, Z body axis input torques, port 3	ft-lbs

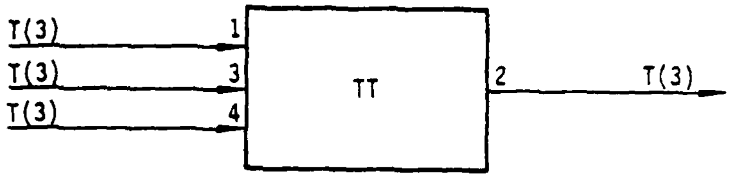
OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
T(3)	2	X, Y, Z body axis output torques, port 2	ft-lbs

EQUATIONS:

$$T2(I) = T1(I) + T3(I) \qquad I = 1, 2, 3$$

SUM THREE SETS OF 3-AXIS TORQUES **TT**



DESCRIPTION: Same as TS, except with one additional port.

US

IUS Vehicle with 6 Degrees of Freedom, Fuel Sloshing, Structural Flexibility, and Tail-wag-dog Engine Dynamics. (This component must be used with component UT to form complete vehicle model.)

INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
DMP(M,M)		Damping matrix (Formed by component UT)	
MSI(M,M)		Inverse Mass Matrix (Formed by component UT)	
WRK(M)		Work vector (Formed by component UT)	
THR(3)		Engine thrust vector in body coordinates	lb
LMN(3)		Spacecraft torque vector due to engine thrust	in-lb
DLM(2)		Moment exerted by actuator on engine nozzle about yaw and pitch axes	in-lb
GNF(N)		Generalized forces due to thrust exerted on flexing modes	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
UVW(3)*		$\begin{pmatrix} u \\ v \\ w \end{pmatrix}$ Rigid body translational velocity vector	
PQR(3)*		$\begin{pmatrix} p \\ q \\ r \end{pmatrix}$ Rigid body rotational velocity vector	
SD1(2)*		$\begin{pmatrix} \dot{S}_{14} \\ \vdots \\ \dot{S}_{10} \end{pmatrix}$ Slosh dynamics velocity vector (1st tank)	
SD2(2)*		$\begin{pmatrix} \dot{S}_{24} \\ \vdots \\ \dot{S}_{20} \end{pmatrix}$ Slosh dynamics velocity vector (2nd tank)	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
DLD(2)*		$\begin{pmatrix} \dot{\delta}_4 \\ \vdots \\ \dot{\delta}_0 \end{pmatrix}$ Nozzle attitude velocity vector	
FXD(N)*		$\begin{pmatrix} \dot{\xi}_1 \\ \vdots \\ \dot{\xi}_n \end{pmatrix}$ Body flex modes velocity vector	
SL1(2)*		$\begin{pmatrix} S_{14} \\ \vdots \\ S_{10} \end{pmatrix}$ Fuel slosh position vector (1st tank)	
SL2(2)*		$\begin{pmatrix} S_{24} \\ \vdots \\ S_{20} \end{pmatrix}$ Fuel slosh position vector (2nd tank)	
DLT(2)*		$\begin{pmatrix} \delta_4 \\ \vdots \\ \delta_0 \end{pmatrix}$ Nozzle attitude vector	
FLX(N)*		$\begin{pmatrix} \xi_1 \\ \vdots \\ \xi_n \end{pmatrix}$ Body flex mode position vector	

EQUATIONS:

$$\underline{\ddot{X}} = \underline{MSI}\cdot\left[-\underline{DMP}\;\underline{\dot{X}} - \underline{STF}\;\underline{X} + \underline{f}\right]$$

MSI, DMP, and STF are M x M Matrices formed by standard component UT.

* These quantities are continuous states.

N must be specified as the number of structural flexibility modes

M must be specified as 12 + N

US

$$
X = \begin{bmatrix} u \\ v \\ w \\ \hline p \\ q \\ r \\ \hline \dot{S}_{14} \\ \vdots \\ \dot{S}_{10} \\ \dot{S}_{24} \\ \hline \vdots \\ \dot{S}_{20} \\ \dot{\delta}_4 \\ \vdots \\ \dot{\delta}_0 \\ \hline \dot{\xi}_1 \\ \vdots \\ \dot{\xi}_n \end{bmatrix}
\begin{array}{l} \left.\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\right\} 12 \\ \left.\begin{array}{c} \\ \\ \\ \end{array}\right\} N \end{array}
\qquad \text{and } f = \begin{bmatrix} THR(1) \\ THR(2) \\ THR(3) \\ \hline L \\ M \\ N \\ \hline 0 \\ \\ 0 \\ 0 \\ \hline 0 \\ DLM(1) \\ \\ DLM(2) \\ \hline GNF(1) \\ \\ GNF(N) \end{bmatrix}
$$

The above represent the spacecraft state vector and the vector of forces due to engine thrust and nozzle actuator, respectively.

This formulation follows Boeing Document D2-84124-4 (pg 70) except that some of the components of X have been permutated for programming convenience.

IUS Vehicle with 6 Degrees of Freedom, Fuel Sloshing, **UT**

Structural Flexibility, and Tail-wag-dog Engine Dynamics.

(This component must be used with component US to form complete Vehicle Model.)

INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
PQR(3)		$\begin{pmatrix} p \\ q \\ r \end{pmatrix}$ Rigid body rotational velocity vector	rad/sec
MS1 MS2		$\left.\begin{matrix} m_{s1} \\ m_{s2} \end{matrix}\right\}$ Sloshing masses for tanks 1 and 2	$lb\text{-}sec^2/in$
LS1 LS2		$\left.\begin{matrix} l_{s1} \\ l_{s2} \end{matrix}\right\}$ Sloshing pendulum arm lengths	inch
SP1(3) SP2(3)		$\left.\begin{matrix} \begin{pmatrix} x_{s1} \\ y_{s1} \\ z_{s1} \end{pmatrix} \\ \begin{pmatrix} x_{s2} \\ y_{s2} \\ z_{s2} \end{pmatrix} \end{matrix}\right\}$ Nominal position of sloshing tanks 1 and 2 in body coordinates	
ME		m_e Mass of engine nozzle	
LE		l Distance from hinge point to nozzle center of gravity	inch
EP(3)		$\begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix}$ Position of nozzle center of gravity in body coordinates when nozzle is in undeflated position	inch
MSS		M Mass of entire spacecraft	$lb\text{-}sec^2/in$
IXX IYY IZZ		$\left.\begin{matrix} I_x \\ I_y \\ I_z \end{matrix}\right\}$ Spacecraft moments of inertia	$in\text{-}lb\text{-}sec^2$
IYE IZE		$\left.\begin{matrix} I_{ye} \\ I_{ze} \end{matrix}\right\}$ Nozzle moments of inertia about nozzle center of gravity	$in\text{-}lb\text{-}sec^2$

INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
IXY IXZ IYZ		I_{xy}, I_{xz}, I_{yz} Spacecraft products of inertia	in-lb-sec^2
MM(N)		$\begin{pmatrix} M_1 \\ \vdots \\ M_n \end{pmatrix}$	
PS1(2,N) PS2(2,N)		$\varphi_{s1}(2,N)$, $\varphi_{s2}(2,N)$ Flex deflection coefficients at tanks 1 and 2	
PE(2,N)		$\varphi_e(2,N)$ Flex deflection coefficients at nozzle	
PEP(2,N)		$\varphi_e(2,N)$ Flex rotation coefficients at nozzle	
WP1 WT1 WP2 WT2		ω_{s14}, ω_{s10}, ω_{s24}, ω_{s20} Natural frequencies at sloshing modes for tanks 1 and 2 about yaw and pitch axes	rad/sec
WFX(N)		$\begin{pmatrix} \omega_1 \\ \vdots \\ \omega_n \end{pmatrix}$ Natural frequencies of flex modes 1, ...,n	rad/sec
WEP WET		ω_{s4}, ω_{s0} Natural frequencies of nozzle in yaw and pitch axes	
ZS1 ZS2		ζ_{s1}, ζ_{s2} Damping ratios of sloshing modes	
ZFX(N)		$\begin{pmatrix} \zeta_1 \\ \vdots \\ \zeta_n \end{pmatrix}$ Damping ratios of flexing modes	
ZEP ZET		ζ_{s4}, ζ_{s0} Linear damping ratio for nozzle about yaw and pitch axes	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION.	UNITS
DMP(M,M)		Damping matrix	
MSI(M,M)		Inverse mass matrix	
STF(M,M)		Stiffness matrix	

EQUATIONS:

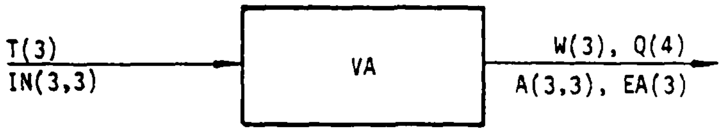
See document D2-84124-4, page 70.

N must be specified as the number of structural flexibility modes

M must be specified as 12 + N

VEHICLE ATTITUDE

VA



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
*LA	1	Initial latitude	deg
*LO	1	Initial longitude	deg
**TI	1	Initial time	hours
***DA	1	Initial date - Julian days	days
*ROL		Initial roll - relative to local horizontal axes	deg
*PIT		Initial pitch - relative to local horizontal axes	deg
*YAW		Initial yaw - relative to local horizontal axes	deg
T(3)		External torques, body axes	ft-lb
IN(3,3)		Inertia matrix, body axes	slug-ft^2

* Default values of zero are provided for these quantities

** Default value of 12 is provided for TI

*** Default value of 80 is provided for DA

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
W(3)		Angular rates - body axes	deg/sec
Q(4)		Quaternians - inertial to body axes	
A(3,3)		Direction cosine matrix - inertial to body axes	
EA(3)		Euler angle - inertial to body axes	deg
LA	2	Initial latitude	deg
LO	2	Initial longitude	deg
TI	2	Initial time	hours
DA	2	Initial date - Julian days	days

VOLTAGE REGULATOR **V6**

INPUT

PHYSICAL QUANTITY	FIGURE	DESCRIPTION	UNITS
VPF	VPF	INPUT FROM POWER FACTOR CONTROLLER	PER UNIT
VL	VL	LINE VOLTAGE	PER UNIT
ED	ED	D AXIS VOLTAGE FROM GEN	PER UNIT
EQ	EQ	Q AXIS VOLTAGE FROM GEN	PER UNIT
VRE	VRE	VOLTAGE REFERENCE	PER UNIT
G1	G1	LAG GAIN	-
G2	G2	LEAD LAG GAIN (FEEDBACK)	-
K	K	FEEDBACK GAIN	-
T1	T1	LAG TIME CONSTANT	-
T2	T2	LEAD LAG TIME CONSTANT (FEEDBACK)	SEC
T3	T3	LEAD LAG TIME CONSTANT (FEEDBACK)	SEC
T4	T4	LAG TIME CONSTANT	SEC
CEX	CEX	LIMITER MAX	SEC
EB	EB	LAG GAIN (PER UNIT CONVERSION)	PER UNIT
G3	G3	SATURATION SLOPE	-

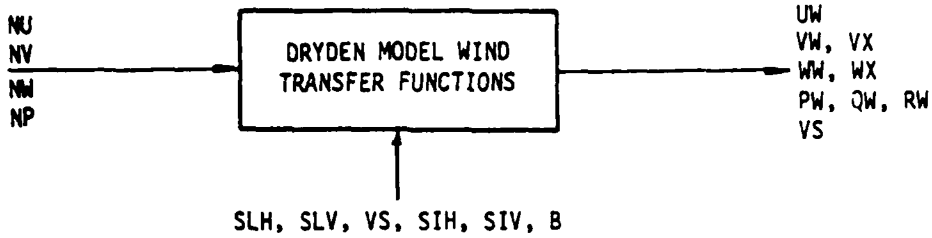
OUTPUT

PHYSICAL QUANTITY NAME	FIGURE NAME	DESCRIPTION	UNITS
* E2	E2	INTERNAL STATE LAG OUTPUT	PER UNIT
* E4	E4	INTERNAL STATE	PER UNIT
* E5		INTERMEDIATE STATE	
* VO	VO	OUTPUT TO GEN/EXCITER	VOLTS
EL	EL	RSS OF EQ AND ED	PER UNIT
E1	E1	ERROR SUM	PER UNIT
E3	E3	LIMITER OUTPUT	PER UNIT

* THESE OUTPUT QUANTITIES ARE STATES

RANDOM WIND GUST MODEL

WM



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
NU, NV, NW		Random noise inputs for UW, VW, WW	
NP		Random noise input for PW angular rate	
SLH, SLV		Horizontal and vertical scales*	ft
VS	1	Steady state airspeed input	ft/sec
SIH, SIV		Horizontal and vertical RMS gust intensity*	ft/sec
B		Wing span	ft

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
UW, VW, WW		X, Y, Z body axis wind velocity states	ft/sec
VX, WX		Y, Z axis intermediate states	ft/sec^2
QX, RX		Y, Z body axis wind angular rate states	deg/sec
PW, QW, RW		X, Y, Z body axis wind angular rate outputs	deg/sec
VS	2	Steady state airspeed	ft/sec

*Default values:

$SLH = SLV = 1750$

$SIH = SIV = 0$

in general, choose SIH and SIV such that $\frac{(SIH)^2}{SLH} = \frac{(SIV)^2}{SLV}$

WIND MODEL TRANSFER EQUATIONS

UW:

$$NU \longrightarrow \boxed{\frac{G_u}{1 + L'_H \cdot S}} \longrightarrow UW$$

$L'_H = SLH/VS$

$G_u = SIH(2 \cdot L'_H/\pi)^{1/2}$

$\dot{UW} = (G_u \cdot NU - UW)/L'_H$

VW:

$$NV \longrightarrow \boxed{\frac{G_v(1 + 3 \cdot L'_H \cdot S)}{(1 + L'_H \cdot S)^2}} \longrightarrow VW$$

$G_v = SIH \cdot (L'_H/\pi)^{1/2} = G_u/\sqrt{2}$

$\dot{VX} = (G_v \cdot NV - VW)/(L'_H)^2$

$VW = VX + (\sqrt{3} \cdot G_v \cdot NV - 2 \cdot VW)/L'_H$

WW:

$$NW \longrightarrow \boxed{\frac{G_w(1 + \sqrt{3} \cdot L'_v \cdot S)}{(1 + L'_v \cdot S)^2}} \longrightarrow WW$$

$L_v = SLV/VS$

$G_w = SIV \cdot (L'_v/\pi)^{1/2}$

$WX = (G_w \cdot NW - WW)/(L'_v)^2$

$WW = WX + (\sqrt{3} \cdot G_w \cdot NW - 2 \cdot WW)/L'_v$

PW:

$$NP \longrightarrow \boxed{\frac{G_p \ (180/\pi)}{1 + C_H \cdot S}} \longrightarrow PW$$

$C_H = 4 \cdot B/(\pi \cdot VS)$

$G_p = SIV \cdot (0.8(\pi \cdot SLV/(4 \cdot B))^{1/3}/(SLV \cdot VS))^{1/2}$

$\dot{PW} = ((GP \cdot NP - PW/C_H) \ 180/\pi$

RW:

$$VW \longrightarrow \boxed{\frac{-S(180/\pi)}{VS(1 + C_V \cdot S)}} \longrightarrow RW$$

$C_V = 3 \cdot B/(\pi \cdot VS) = .75 \cdot C_H$

$RW = RX - 180/\pi \cdot VW/(VS \cdot C_V)$

$\dot{RX} = -RW/C_V$

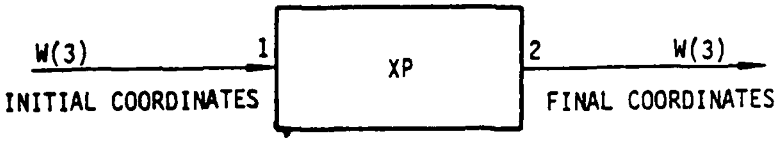
QW:

$$WW \longrightarrow \boxed{\frac{S(180/\pi)}{VS(1 + C_H \cdot S)}} \longrightarrow QW$$

$QW = QX + 180/\pi \cdot WW/(VS \cdot C_H)$

$\dot{QX} = -QW/C_H$

STATIC TRANSFORMATION OF ANGULAR RATES **XP**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
W(3)	1	Input angular rates - initial coordinates	rad/sec
TRN(3,3)		3 x 3 transformation matrix	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
W(3)	2	Output angular rates - final coordinates	rad/sec

EQUATIONS:

W2 = TRN•W1 (Matrix Multiply)

ASSUMPTIONS:

TRN contains the direction cosines required to transform from the initial coordinate system. TRN is input as follows:

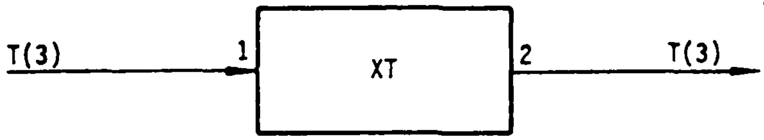
PARAMETER VALUES = TRNXP

R(1,1) a_{11}, a_{12}, a_{13}

R(2,1) a_{21}, a_{22}, a_{23}

R(3,1) a_{31}, a_{32}, a_{33}

STATIC TRANSFORMATION OF TORQUES **XT**



INPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
T(3)	1	Input torques - initial coordinates	ft-lbs
TRN		3 x 3 transformation matrix	

OUTPUT

PHYSICAL QUANTITY NAME	PORT NO.	DESCRIPTION	UNITS
T(3)	2	Output torques - final coordinates	ft-lbs

APPENDIX L

EASY PROGRAM ANALYSIS DESCRIPTION

This appendix is a reproduction of Section 4.4 of reference 1. It presents a description of the mathematical methods used in each of the anlayses available in the EASY Analysis Program.

4.3.2 Scalar Data

Scalar data, i.e., parameter values, error controls and initial conditions, should be loaded by data cards immediately following any tabular data cards. All of these scalar values should be specified before any analysis is requested. However, to prevent the loss of an analysis run due to the omission of one or more parameter values, error controls, or initial conditions, all parameter values are initialized to a default value of 0.99999, all error controls to 0.1*, and all inital conditions to 0. Sections 4.2.2 and 4.2.3 describe the program commands and formats used to specify scalar data. Once the parameter values, error controls, and initial conditions have been specified, other program commands may be issued to request analyses. The values of any of the scalar data can be modified between analysis requests by using the same commands described in Sections 4.2.2 and 4.2.3.

4.4 ANALYSIS DESCRIPTIONS

This section contains a description of the mathematical methods used in each of the analyses available in the EASY Analysis Program. Further details of each analysis can be found in the Section 6.

4.4.1 Simulation Calculations

One of the most used and well known numerical integration rules is the classical explicit fourth order Runge-Kutta method (Reference 1). The method is easy to implement, has nice truncation error properties, and combined with an error control (step size adjustment) is a good standard integration method for systems with eigenvalues (of the Jacobian) all relatively the same size. For this reason, the 4th order Runge-

---

* See Section 4.5.3 for special default values provided by the EASY Model Generation program for states whose name starts with the letters P or T.

Kutta method is included as one integrator available in EASY. It is not the default integrator, however, because of its stability properties. A short discussion of integration rule stability follows.

For most integration algorithms, truncation error (the error incurred due to a finite order approximation to the exact solution) is directly related to the step size raised to a power equal to the order of the method. By controlling the step size, the single step error can theoretically be maintained at any desired level. This assumes that sufficient precision is used so that round off effects (error due to approximating numbers by a finite number of bits or digits) does not become a factor. Most integration algorithms thus contain some error measurement calculation and a step size adjuster so that single step error is below a specified limit.

The question now arises of what happens to such systems when the actual value of the truncation error becomes very small due to the actual solution approaching a steady or slowly varying value. The normal logic in most algorithms indicates that the step size should be increased. As the step size is increased a phenomenon related to integration rule stability occurs. That is, even though the solution and the resultant error are well below the specified error limit, increasing the step size will eventually cause errors to increase over the limit. This is due to the fact that every integration rule has a region of stability (a function of step size) where given a stable (non increasing) system it will compute a nonincreasing solution. Outside of that region, even though the solution should decrease, it will compute an increasing solution. This region is normally described as a function of the time step h times the complex eigenvalues of the system. Thus if one were to plot the region in a complex plane modified by the step size, the 4th order Runge-Kutta would have a region that appears as shown in Figure 42.

This means that if any stable mode (represented by a eigenvalue λ_i) is large enough that $h\lambda_i$ lies outside the shaded area, then for that

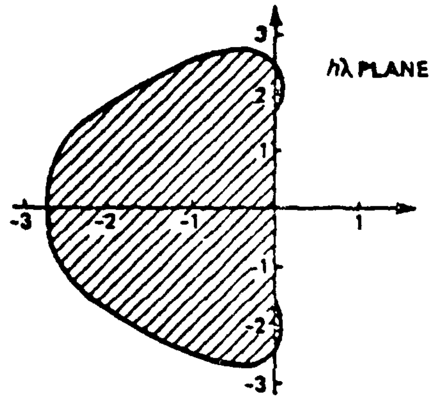


Figure 42. Region of Absolute Stability of Fourth Order Explicit Runge-Kutta Method

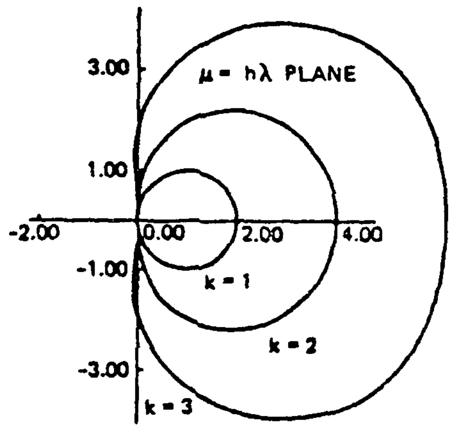


Figure 43. Regions of Absolute Stability for Stiffly Stable Methods of Orders One Through Three. Methods Are Stable Outside of Closed Contours.

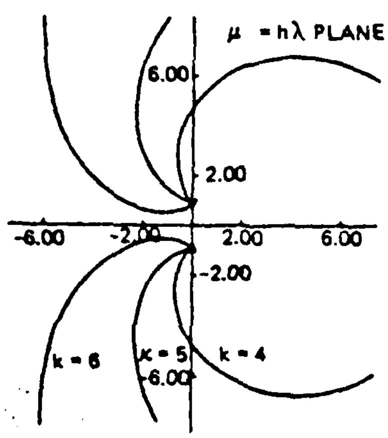


Figure 44. Regions of Absolute Stability for Stiffly Stable Methods of Orders Four Through Six

mode and step size h, an increasing solution will be computed even though the actual solution is decreasing. For this reason, even when truncation error is reduced to a very small value and the solution mode is in steady state, the step size for the Runge-Kutta method is limited to approximately

$$h_{max} < \frac{2.7}{|\lambda_{max}|}$$

in order to prevent the computed solution from diverging.

For systems that have wide ranges of eigenvalues, this limitation can cause unreasonably long computation times. Thus, one seeks integration rules which have more desirable stability properties.

The integration rule implemented in EASY is a "stiffly stable" method developed by Gear and published both in his book and in the Communications of the Association for Computing Machinery, Vol. 14, No. 3, March 1971. This is a variable step size, variable order method which has regions of stability outside of the contours in Figures 43 and 44. For these regions, it is noted that large magnitude eigenvalues with negative real parts that are large fall well inside the region of stability. Thus as truncation error becomes small during the integration process, the method is not restricted from using large step sizes.

Note that part of the right hand plane is stable even though the actual system would be unstable. All this means is that if the step size were unchanged, the integrator would output a decreasing solution. Error control, however, would detect the difference and decrease the step size until a correct solution to the specified accuracy was obtained.

Since the algorithm is well documented in Gear's book in Chapter 9, the theoretical exposition is not repeated here. The modifications made to the data structure so that storage is by column and not by row. Theoretically this is of no importance but practically it is better due to the manner in which FORTRAN stores and computes indices in arrays.

Only the stiff integrator that computes partial derivatives by numerical differencing is retained. The process of solving a linear system of equations by matrix inversion is replaced by the more efficient and accurate direct Gaussian elimination method. The method order is restricted to 5 or less because of stability considerations.

The process of integration is controlled by a master subroutine which keeps track of time and the necessary reporting sequence. Further, this routine recognizes when a new call is made to the integrator for the first time and uses a special start up procedure. This procedure essentially uses the standard 4th order Runge-Kutta for 100 steps (picked by the step size controller) to let initial transients settle out before handing the problem over to Gear's method. Since the Gear method must start out with a 1st order integration rule, large initial transients can cause problems. Thus using another 4th order rule to integrate over small intervals of large transient behavior allows the Gear method to start in a smoother region of the solution. This external integration process will occur whenever large transients cause the Gear method to fail.

Minimum step size is set at 10^{-5} seconds or TINC/10000, whichever is the smaller value. The maximum step size is set equal to the print interval and is often attained. The error test used is based on relative error with respect to the maximum value computed for a particular variable. The current value is set at 5 significant figures maintained over a single step.

At the start of each simulation run, the time variable is set equal to zero; the state vector of the system model is set equal to the initial condition vector, (values input via the INITIAL CONDITION command); and the state variable time derivatives (rates), are set equal to zero. The rates are set equal to zero as part of the procedure that allows individual states to be frozen.

For frozen states, the rates are not recalculated by the system model. Thus, since the rates are set to zero these states remain "frozen" at their initial values.

Integration of the system model equations continues until the value of time equals the value of TMAX specified by the analyst. If it is desired to have a simulation stop for some condition, before time reaches TMAX, a test on this condition can be added to the system model, (in subroutine EQMO), and TIME set equal to TMAX should this condition occur. An example of this sort was shown in Example 3.3.

4.4.2 Steady State Calculations

The STEADY STATE option allows the steady state of a stable system dynamic model to be quickly determined. This is accomplished by modifying the dynamic characteristics of the system so that all eigenvalues are near -1. This allows the system transient to be quickly integrated to reach steady state.

The nonlinear simulation model can be defined as:

$$\underline{\dot{x}} = \underline{f}\,(\underline{x},t) \qquad 4.4\text{-}1$$

where: $\underline{\dot{x}}$ = n dimensional vector of state variable derivatives

$\underline{x}$ = n dimensional vector of state variables

$\underline{f}$ = n dimensional vector of nonlinear functions relating state variables and time to state variable derivatives.

The steady state of this system is defined as that value, $\underline{x}_{ss}$, of the system state vector, $\underline{x}$, that causes $\underline{\dot{x}}$ to equal zero. Thus:

$$\underline{0} = \underline{f}\,(\underline{x}_{ss},\, t) \qquad 4.4\text{-}2$$

Let a linear approximation for the nonlinear system, as described in Section 4.5.3, be given by:

$$\underline{\dot{x}} = \underline{Ax} \qquad 4.4\text{-}3$$

Where $\underline{A}$ = nxn stability matrix (Jacobian) of the system model.

The major objection to integrating the given nonlinear system of (4.4-1) to obtain the steady state is that many small integration steps are required over a long transient duration to reach steady state. As discussed in Section 4.4.1 this problem is related to a large range of eignevalue magnitudes of the system stability matrix, $\underline{A}$. If the objective is to rapidly reach steady state, the ideal dynamic system would have all of its eigenvalues concentrated in a very small range. This can be accomplished, if one is not interested in the accuracy of the transient calculation, for a stable system with a negative definite $\underline{A}$ by premultiplying the system matrix by $-\underline{A}^{-1}$. The modified state will be designated by $\underline{x}'$.

$$\dot{\underline{x}}' = -\underline{A}^{-1} \underline{A}\, \underline{x}' \qquad 4.4\text{-}4$$

$$= -\underline{I}\, \underline{x}' \qquad 4.4\text{-}5$$

The modified system of equation (4.4-5) has the desired feature that all of its eigenvalues are in a small range, i.e., all equal minus one. Thus, by pre-multiplying the given system function by $-\underline{A}^{-1}$, we may obtain a modified system with all eigenvalues near -1. Applying this modification to equation 4.4-1 we obtain

$$\dot{\underline{x}}' = -\underline{A}^{-1} \underline{f}\, (\underline{x}', \underline{t}) \qquad 4.4\text{-}6$$

Since the transformation $\underline{A}^{-1}$ is nonsingular, the only solution to the modified steady state equation

$$\underline{0} = -\underline{A}^{-1}\underline{f}\, (\underline{x}_{ss}, t) \qquad 4.4\text{-}7$$

is that shown in equation (4.4-2). Thus the system of equations given in (4.4-6) has the same steady state solution as the original system, (4.4-1) but has an eigenvalue range that greatly reduces the number of integration steps required to reach steady state. This approach to solving for the steady state may also be viewed as a multi-dimensional

version of Newton's Method for solving the nonlinear algebraic equation of (4.4-2). The numerical method proceeds as follows:

The system rates and stability matrix are evaluated at the initial state, $\underline{x}_i$.

$$\dot{\underline{x}}_i = \underline{f}\,(\underline{x}_i,\ \underline{t}) \qquad 4.4\text{-}8$$

$$\underline{A}_i = \left.\frac{\partial \underline{f}(\underline{x},t)}{\partial \underline{x}}\right|_{\underline{x}=x_i} \qquad 4.4\text{-}9$$

Rather than premultiply by the inverse matrix, as indicated in (4.4-6), the equation

$$-\underline{A}_i\ \dot{\underline{x}}_i = \underline{f}\,(\underline{x}_i,\ \underline{t}). \qquad 4.4\text{-}10$$

is solved for $\dot{\underline{x}}_i$, given $\underline{A}_i$ and $\underline{f}(\underline{x}_i,t)$ by the Gaussian elimination method.

The Euler forward difference approximation, for a time difference of 1, is then used to represent $\dot{\underline{x}}_i$

$$\dot{\underline{x}}_i = \underline{x}_{i+1} - \underline{x}_i \qquad 4.4\text{-}11$$

Solving for $\underline{x}_{i+1}$ we obtain

$$\underline{x}_{i+1} = \underline{x}_i + \dot{\underline{x}}_i \qquad 4.4\text{-}12$$

The process of solving equations (4.4-8) through (4.4-12) is repeated until the norm of the residual vector, $\dot{\underline{x}}$, becomes less than 10^{-4} or more than SS ITERATIONS occur. As implemented, the system stability matrix $\underline{A}$ is not completely recalculated each iteration and a step size less than 1 second is used if the method encounters difficulty in converging.

Should this method fail to reach a steady state from a given initial condition, the less efficient, but more stable simulation approach can be used. Of course, for some nonlinear systems a steady state can not be reached from certain regions of the state space, (initial conditions). In these cases, it will be necessary to vary the initial conditions to find a steady state by either the STEADY STATE, or the SIMULATE commands.

At the final state reached by the steady state analysis, a linear model of the system is generated and its eigenvalues are calculated and printed. These should be examined to assure that there are no non-negative real parts which would indicate an unstable system. It is usually of interest to know the eigenvalues of the system at each steady state operating point. Also, in rare cases, the steady state method can converge to an unstable equilibrium point such as point X2 in Figure 45.

4.4.3 Linear Analysis Calculations

Stability Matrix Calculation

The LINEAR ANALYSIS option allows linear approximations to the nonlinear system model to be generated at any given operating point. This analysis calculates the stability matrix (i.e. Jacobian) of the nonlinear system model and the eigenvalues of that matrix. This analysis can be described as follows. The nonlinear system model can be defined as:

$$\underline{\dot{x}} = \underline{f}(\underline{x},t) \tag{4.4-13}$$

where: $\underline{\dot{x}}$ = n dimensional vector of state variable derivatives

$\underline{x}$ = n dimensional vector of state variables

$\underline{f}$ = n dimensional vector of nonlinear functions relating state variables and time to state variable derivatives.

A linear model of this nonlinear system can be expressed as:

$$\underline{\dot{x}} = \underline{A}\ \underline{x} \tag{4.4-14}$$

where $\underline{A}$ = n x n system stability matrix

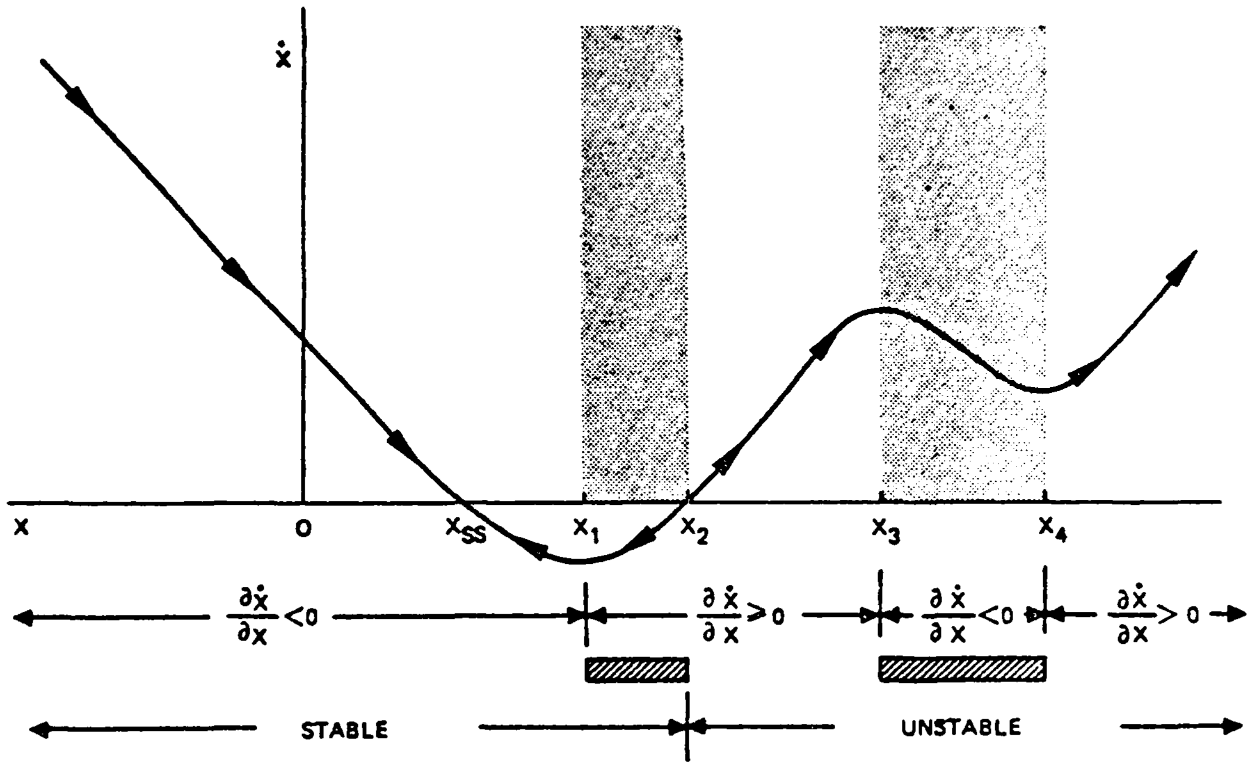


Figure 45. Nonlinear Stability Example

The ij^{th} element of $\underline{A}$, a_{ij}, is related to the partial derivative of the elements of $\underline{f}$ with respect to the elements of $\underline{x}$, at the operating point $\underline{x}_0$ as

$$a_{ij} = \left. \frac{\partial f_i\ (\underline{x},t)}{\partial x_j} \right|_{\underline{x} = \underline{x}_0} \qquad 4.4\text{-}15$$

The eigenvalues of the stability matrix are a set of n complex numbers that characterize the dynamic behavior of the system in a region about the chosen operating point, $\underline{x}_0$. Eigenvalues with non-negative real parts indicate that the system is unstable in the region about, $\underline{x}_0$.

It must be kept in mind that for highly nonlinear systems, this simple measure of stability is not a necessary or sufficient condition for stable operation. This can be demonstrated with a simple first order system as shown in Figure 45. For this example, the state derivative $\dot{x}$ is shown as a highly nonlinear function of the single system state variable, x. The arrows on the plot of the function show the trajectory the state, and state derivative would follow from any initial state x. For the values of x shown, there is a stable region, and an unstable region. Initial values of x in the stable region will result in the system reaching the steady state operating point, x_{ss}. Initial values of $x>x_2$ will result in x diverging to large positive values. The eigenvalue of this simple system is the partial derivative, $\frac{\partial \dot{x}}{\partial x}$. We see that the simple criteria of a negative real eigenvalue for stability specifies that the system is unstable in the region x_1 to x_2, while for this example, it will converge to the steady state point, x_{ss}. In the region x_3 to x_4 the eigenvalue criteria would indicate that the system was stable, while in fact it will diverge from this region.

This example is presented to illustrate the hazards that exist when using eigenvalues to measure system stability at points other than steady state operating points. However, much useful information and

insights into system behavior can be obtained from such linear analyses. Especially since they can be easily verified by the nonlinear simulation capabilities of the EASY Analysis program.

The numerical method used to calculate the stability matrix is as follows: The values of the state variable derivatives, (rates) are calculated at the given operating point, $\underline{x}_o$

$$\dot{x} = \underline{f}\,(\underline{x}_0, 0) \qquad 4.4\text{-}16$$

where: $\dot{\underline{x}}_0$ = n dimensional vector of state derivatives at operating point

$\underline{x}_0$ = n dimensional vector of state variables which specifies the operating point.

$\underline{f}$ = n dimensional vector of nonlinear functions relating state variables to state derivatives.

These values are printed and should be examined to determine if the operating point is a steady state operating point, i.e. ($\dot{\underline{x}}_0 = 0$).

Non zero elements of $\dot{\underline{x}}_0$, (rates), indicate the sign and magnitude of unbalance at the chosen operating point.

The j^{th} element of the operating point vector is perturbed by adding the j^{th} element of the error vector, e_j.* This perturbed operating point is used to recalculate the state variable derivatives, $\dot{\underline{x}}_j$. The j^{th} column of the stability matrix, $\underline{A}_j$, is then calculated as:

$$\{\underline{A}\}_j = \frac{\dot{\underline{x}}_j - \dot{\underline{x}}_o}{e_j} \qquad 4.4\text{-}17$$

* Note: this is the same vector that is used for integration error control. It's values are furnished to the program via the ERROR CONTROL commands.

where: $\dot{\underline{x}}_j$ = n dimensional vector of state derivatives at the operating point, perturbed by adding j^{th} element of error vector to $\underline{x}_o$.

e_j = j^{th} element of the error vector.

$\{\underline{A}\}_j$ = j^{th} column of the system stability matrix.

This process is repeated for all n columns of $\underline{A}$.

As a measure of the validity of the linear approximation, the stability matrix calculation described above is repeated using perturbations one half those used in the initial calculation.

The ratios of the derivatives calculated with the two step sizes are evaluated and placed in an array, RATIO. If the results of measuring all derivatives with both step sizes are equal, all elements of RATIO will equal one.

The elements of RATIO are compared to one and the number of elements differing from one by more than ten percent noted. If one or more such elements is found, the count of such elements is recorded on the printer along with a list of the elements of RATIO that exceed the tolerance of ten percent.

Figure 46 shows an example of how the values in the array RATIO may be used to measure the local linearity of the system model.

Eigenvalue Calculation

The method used to compute the eigenvalues of the system stability matrix consists of three basic steps. The first step is the conditioning of the matrix prior to the application of the normal transformation process. The conditioning process is divided into two steps of reduction and scaling. Reduction is the process whereby through row and column interchange the matrix is transformed into upper block triangular form. This means that the diagonal blocks can be treated independently for the

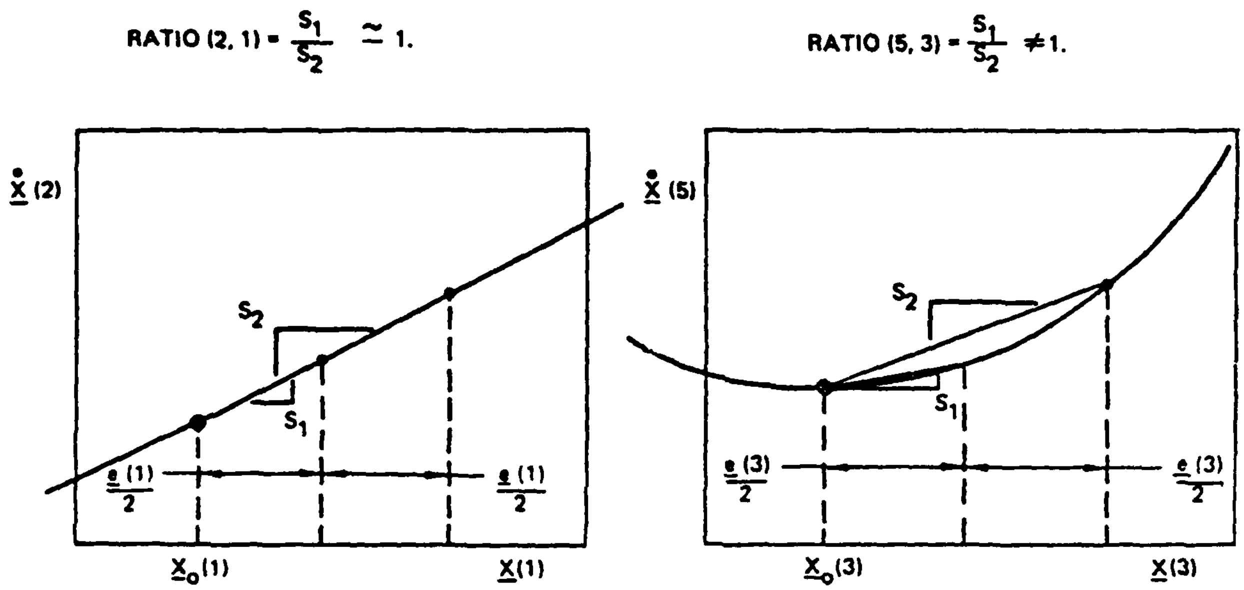


Figure 46. Linearity Measure Example

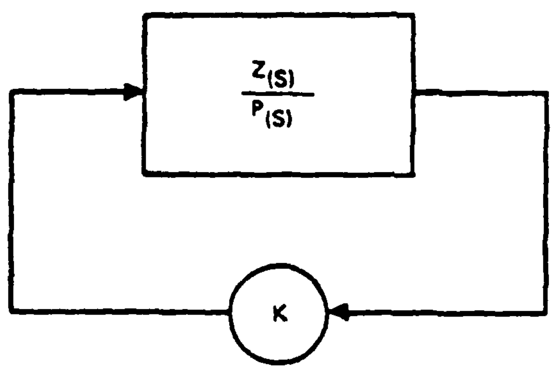


Figure 47. Equivalent Stability Margin System

purpose of eigenvalue calculation. This reduction naturally occurs whenever openloop or feed forward systems are described. The algorithm used for reduction is described in detail in Appendix A under the title of the McCreight algorithm. For hisotrical perspective, an earlier method proposed by Harary is given. The second phase of the conditioning process is scaling. Since the errors in all the transformation algorithms used subsequent to the conditioning process are related to the norm of the matrix, scaling is used to reduce the norm. Historically it was though that the need for scaling was eliminated when the transition form analog to digital computers was made. Modern numerical analysis indicates that this is not true and that proper scaling is important to minimize loss of significance in computed results. The scaling algorithm used is one developed by E. E. Osborne in 1960 and consists of a sequence of diagonal transformations to minimize the Euclidean norm of a irreducible matrix. Since the reduction process was performed first, each diagonal block is irreducible and the scaling algorithm applies. Details of the algorithm are explained in Appendix A.

The second process in the computing of eigenvalues is to transform the scaled diagonal blocks determined in the first step into upper Hessenburg form. This form, where all the elements below the first sub-diagonal are zero, is most convenient and efficient for further calculation. In Appedix A, two methods are discussed with the "direct reduction with interchanges" being the method implemented.

The final step in the computation of eigenvalues is the actual determination of the eigenvalues for each diagonal block (now scaled and in Hessenburg form). The algorithm used is the QR algorithm developed by Francis in the early 1960's and described in Appendix A. The algorithm uses a series of unitary transformations to drive elements of the subdiagonal of the Hessenburg form to effective zero values. As the subdiagonal elements approach zero, the diagonal elements approach the

the desired eigenvalues. The algorithm is very efficient and quite suitable for problems of moderate size (less than 100-200 order).

Appendix A, which is comprised of notes from a series of lectures, presents the basic mathematics of each of the above processes along with numerical examples to demonstrate the actual computing sequence.

4.4.4 Stability Margin Calculations

The method that is used to determine stability margins is a frequency domain technique of Bode. This technique has been found to be numerically superior to other approaches, such as the Routh array approach and much faster than the direct approach of repeated eigenvalues determination.

The parameter K for which the stability margin is to be calculated can be though of as providing a single loop feeback around the system model as shown in Figure 47.

The characteristic equation of the above system with nominal parameter $K = K_n$, is:

$$N(s) = P(s) - K_n Z(s) \qquad 4.4\text{-}18$$

Note that the sign of the feeback is determined by the sign of K and is not assumed to be negative as is often the case in text books. The roots of N(s) are the eigenvalues of the nominal system, and the roots of P(s) are the eigenvalues of the system with K = 0.

To concentrate the analysis on the stability boundary of the complex plane, i.e. the imaginary axis, we may set $s = j\omega$ in Equation 4.4-18. The polynomials $P(j\omega)$ become complex quantities for real values of ω.

We are interested in determining those real values of K, K_o, which will cause $N(j\omega) = 0$. Such values of K will result in roots of the characteristic equation on the imaginary axis of the complex plane.

Solving equation (4.4-18) for such values of K we obtain:

$$0 = P(j\omega) - K_0 Z(j\omega) \qquad 4.4\text{-}19$$

$$K_0 = \frac{P(j\omega)}{Z(j\omega)} \qquad 4.4\text{-}20$$

Since we are interested in only real values of K_0 that satisfy 4.4-19, we need consider only those values of ω which cause the phase of $P(j\omega)/Z(j\omega)$ to equal 0^0 or 180^0. Further, if the nominal parameter $K_n < 0$, only values of 180^0 need to be considered, and if the nominal parameter $K_n > 0$, only the values of ω that produces 0^0 phase need be considered.

The approach that will be taken to determine K_0 will be as follows. The roots of N(s) and P(s) of 4.4-18 can be calculated as the eigenvalues of the nominal system, and the eigenvalues with K = 0 respectively, and will be designated as:

$N_i \qquad i = 1,2,\ldots\ldots, n \quad K = K_n$

$P_i \qquad i = 1,2,\ldots\ldots, n \quad K = 0$

Thus N(s) and P(s) can be stated in terms of their roots as:

$$P(s) = \prod_{i=1}^{n} (s - P_i)$$

$$N(s) = \prod_{i=1} (s - N_i) \qquad 4.4\text{-}22$$

Solving 4.4-18 for the open loop transfer function in terms of K_n, N(s) and P(s) we obtain:

$$\frac{Z(s)}{P(s)} = \frac{1}{K_n}\left[1 - \frac{N(s)}{P(s)}\right]$$

4.4-23

$$= \frac{R(s)}{K_n}$$

Where:

$$R(s) = 1 - \frac{N(s)}{P(s)}$$

If $K_n > 0$, the phase of $\frac{Z(s)}{P(s)}$ is the phase of R(s). If $K_n < 0$, the phase of $\frac{Z(s)}{R(s)}$ is the phase of R(s) minus 180^o. Thus, the method simplifies to a search for the frequencies that cause the phase of R(s) to be 0^o, regardless of the sign of K_n.

Substituting $s = j\omega$ into 4.4-22 and 4.4-22 into 4.4-23 we obtain

$$\frac{Z\ (j\omega)}{P\ (j\omega)} = \frac{1}{K_n}\left[1 = \prod_{i=1}^{n} \frac{(j\omega - N_i)}{(j\omega - P_i)}\right] \qquad 4.4\text{-}24$$

$$= \frac{R\ (j\omega)}{K_n} \qquad 4.4\text{-}25$$

A range of $0 \leq \omega \leq \omega_{max}$ will be searched to find those values of ω at which the phase of $R(j\omega)$ is zero. At this freqûency, ω_o, the limiting value of K, K_o, can be calculated by substituting 4.4-25 into 4.4-20:

$$K_o = \frac{K_n}{||R(j\omega_o)\ ||} \qquad 4.4\text{-}26$$

Magnitudes of $R(j\omega) > 1.$ result in lower K limits. Magnitudes of $R(j\omega) < 1.$ determine upper K_o limits. The usual definition of stability

margin is the ratio of maximum K, to nominal, K_n is obtained from 4.4-26 to be:

$$\frac{K_o}{K_n} = \frac{1}{R(j\omega_o)} \qquad 4.4\text{-}27$$

Search for Zero Phase

A range of ω from 0 to ω_{max} must be searched for zero crossings of $R(j\omega)$. ω_{max} is arbitrarily established as 2 times the magnitude of the largest eigenvalue of the nominal system. Zero frequency is included since a real divergence is indicated by a zero phase of $R(0)$. After $\omega = 0$ has been checked, the search begins at some low frequency ω_{min}. Since we are interested in phase angles near 0, small angle approximations may be used for the phase of $R(j\omega)$. By this approach it will be possible to avoid time consuming trigonometric calculations. Thus phase angle of $R(j\omega)$ will be approximated as:

$$\angle R(j\omega) \approx \frac{\text{Im } R(j\omega)}{\text{Re } R(j\omega)} \qquad 4.4\text{-}28$$

The search proceeds with geometric steps from ω_{min}. When a zero crossing occurs, the search switches to a dichotomous mode until the error is reduced to some tolerance ε., i.e.

$$|\angle R(j\omega_o)| \leq \varepsilon = .00001 \text{ radian} \qquad 4.4\text{-}29$$

A further condition is included in this search strategy. That is that the phase angles determined on two subsequent geometric search steps should not differ by more than one quadrant. This condition is included to prevent the search from not detecting a zero crossing in a region of rapidly changing phase.

The mode of the search can be easily related to the standard quadrant designations of the phase angles as described below.

The absolute value of the difference of the quadrant numbers of the current and previous phase angle is calculated. If this value is less than two, the geometric search is continued. If this value is equal to two, a small step backward is taken, since a change of two quadrants has occured and a zero crossing may have been overlooked. If this value is greater than two, the phase angle has passed from the first to fourth (or visa-versa), quadrant and a dichotomous search is started to locate the value of frequency that produces zero phase.

When such a value of frequency is determined, the value, ω_0 and the stability margin, $\frac{1}{R(j\omega_0)}$, are stored in arrays, and the search continues in the geometric fashion until ω_{max} is reached.

At this point in the analysis, there are two arrays of k elements $\Omega(i)$ and GM(i) that contain the frequencies ω_0 and the corresponding magnitudes $\frac{1}{R(j\omega_0)}$ respectively. The lower stability limit is determined by the maximum value of $\frac{1}{R(j\omega_0)}$ which is less than 1. The upper stability limit is determined by the minimum value of $\frac{1}{R(j\omega_0)}$ which is greater than 1. The k elements of GM(i) are searched to determine these values. Any remaining elements of GM(i) and $\Omega(i)$ indicate parameter values and divergence frequencies which exceed the critical stability limits, but at which another oscillation would occur if the parameter were increased beyond the critical stability bounds. If such values exist in the searched region, they will be printed out by the program as noncritical stability limits.

4.4.5 Transfer Function Calculations

The method that is used to calculate transfer functions is very similar to that used to calculate stability margins. In each case, the eigen-

values of the nominal system, and the eigenvalues of a related system, are calculated and used to obtain the desired results. Since the eigenvalues of a linearized system can be calculated quite efficiently and accurately, this approach provides an efficient and accurate method of obtaining specified transfer functions.

The transfer function from any point R to any point C in the system model can be represented as shown in Figure 48. The transfer function between points R and C is composed of the ratio of rational polynomials Z(s) and P(s).

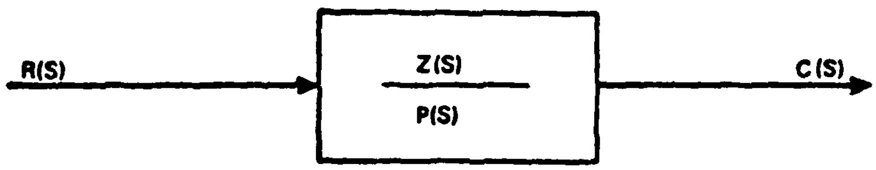
$$\frac{C(s)}{R(s)} = \frac{Z(s)}{P(s)} \qquad 4.4\text{-}30$$

where:
- R(s) - the specified input quantity.
- C(s) - the specified output quantity.
- Z(s) - transfer function numerator polynomial.
- P(s) - transfer function denominator polynomial.
- s - Laplace complex frequency variable.

The roots of the denominator, P(s) can be obtained by forming a linear representation of the system and calculating the nominal system eigenvalues, as discussed in Section 4.4-3. If the equivalent transfer function system of Figure 49 is modified by adding a feedback path from the specified output quantity to the input quantity, we obtain new dynamic system whose transfer function is:

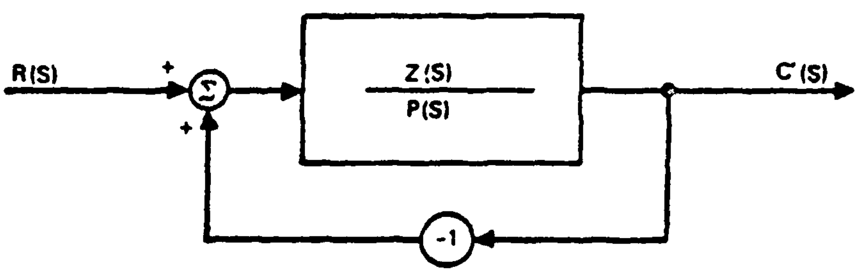
$$\frac{C'(s)}{R(s)} = \frac{Z(s)}{Z(s) + P(s)} \qquad 4.4\text{-}31$$

Let the roots of P(s), the nominal system eigenvalues, be designated as P_i, and the roots of Z(s) + P(s), the modified system eigenvalues, be designated N_i, $i = 1, 2, \ldots, n$ where n is the system order.



$$\frac{C(S)}{R(S)} = \frac{Z(S)}{P(S)}$$

Figure 48. *Equivalent Transfer Function*



$$\frac{C'(S)}{R(S)} = \frac{Z(S)}{Z(S) + P(S)}$$

Figure 49. *Modified Equivalent Transfer Function System*

$$P(s) = \prod_{i=1}^{n} (s-P_i) \qquad 4.4\text{-}32$$

$$Z(s) + P(s) = \prod_{i=1}^{n} (s-N_i) \qquad 4.4\text{-}33$$

The desired transfer function, $\frac{Z(s)}{P(s)}$, can be obtained in terms of the two sets of eigenvalues P_i and N_i by dividing equation 4.4-33 by 4.4-32.

$$\frac{Z(s)}{P(s)} = \frac{\prod_{i=1}^{n} (s-N_i)}{\prod_{i=1}^{n} (s-P_i)} - 1 \qquad 4.4\text{-}34$$

Since we are interested in the steady state frequency response, we will confine our attention to the imaginary axis of the S plane, by replacing s with $j\omega$.

$$\frac{Z(j\omega)}{P(j\omega)} = \frac{\prod_{i=1}^{n} (j\omega-N_i)}{\prod_{i=1}^{n} (j\omega-P_i)} - 1 \qquad 4.4\text{-}35$$

Equation 4.4-35 gives the desired transfer function in terms of the eigenvalues of the nominal system, and that system modified by a single loop closure. Since N_i and P_i are, in general, complex quantities, and the $j\omega$ terms are pure imaginary quantities, the transfer function will be a complex function of ω.

The numerical methods that are used to calculate the nominal system stability matrix and eigenvalues are described in Section 4.4.3. The modified system stability matrix is calculated as follows: First, the nominal value of the specified output quantity, C_o, is determined. At each step of the stability matrix calculation, after a jth state variable has been perturbed, the difference between the resulting value of C, C_j, and the nominal value C_o is subtracted from the current value of the input quantity, R_j.

$$R'_j = R_j - (C_j - C_o) \qquad 4.4\text{-}36$$

where:
- $R'j$ - input quantity modified by -1 loop closure from C.
- Rj - input quantity without -1 loop closure from C.
- C_o - nominal value of output quantity.
- C_i - output quantity value resulting from perturbing jth state variable.

The system model is then re-evaluated from the point in the model equations at which R appears. In this way the effect of a -1 loop closure from output to input is simulated. Note, that this technique fails if the output quantity is a direct, algebraic function of the input quantity. In such a case, the change in C would cause a change in R via (4.4-36), which would cause a further change in C, etc. A test for such "algebraic loops" is performed before the transfer function analysis is allowed to proceed. This situation only occurs in those cases in which the transfer function numerator polynomial and denominator polynomial are of the same order. This situation is fortunately quite uncommon in most physical dynamic systems.

4.4.6 Root Locus Calculation

A root locus analysis provides the locus of the system eigenvalues as a function of some specified parameter. The EASY Analysis program allows a root locus analysis to be performed as a function of any operating

point value, as well as any system parameter.

The root loci are calculated by forming the stability matrix for the system for each specified value of the root locus parameter. The eigenvalues of each stability matrix are calculated to give the root loci.

The methods described in Section 4.4.3 are used to calculate the system stability matrices and eigenvalues. However, the calculation of the linearity measure, RATIO, is omitted for two different* values of the root locus parameter, a comparison of the elements of these stability matrices is made to determine which elements are affected by changes in parameter. Subsequent stability matrix calculations only re-evaluate those elements which were modified by the first two values of the root locus parameter. Due to storage limitations, a limit has been placed on the number of elements that can be modified by the root locus parameter. This limit is 400 elements of the stability matrix. If more than 400 elements of the stability matrix are modified by the root locus parameter, the program reverts to the less efficient process of evaluating all elements of the stability matrix for each value of the root locus parameter.

4.4.7 Eigenvalue Sensitivity Calculations

An eigenvalue sensitivity analysis provides a measure of the sensitivity of system eigenvalues to changes in a specified system parameter. The eigenvalue sensitivity measure is the ratio of the percentage change in the parameter for which the sensitivity is to be measured. This is stated

---

* The two different values are the nominal parameter value and the RL START value. Therefore RL START <u>should not</u> equal the nominal parameter value.

mathematically as:

$$S_{\sigma i} = \frac{1 - \frac{\sigma i'}{\sigma i}}{\left[1 - \frac{P'}{P}\right]} \qquad 4.4\text{-}37$$

$$S_{\omega i} = \frac{1 - \frac{\omega i'}{\omega i}}{\left[1 - \frac{P'}{P}\right]} \qquad 4.4\text{-}38$$

Where: $S_{\sigma i}$ = Sensitivity measure of real part of i^{th} eigenvalue to change in parameter P.

$S_{\omega i}$ = Sensitivity measure of imaginary part of i^{th} eigenvalue to changes in parameter P.

σ_i = Nominal value of real part of i^{th} eigenvalue

ω_i = Nominal value of imaginary part of i^{th} eigenvalue

P = Nominal value of parameter for which sensitivity measure is being calculated

$'$ = Prime indicates perturbed values of parameters and eigenvalues

i = 1,2,...., n n = model order

This sensitivity measure has the following properties:

a. It is dimensionless which allows the relative sensitivities of parameters with different units to be compared.
b. Sensitivity measure of one idicates equal percentage change in eigenvalue per unit change in the parameter.
c. Positive sensitivity indicate eigenvalue motion toward the right half plane, i.e., destabilizing and lower frequencies.
d. Negative sensitivities indicate eigenvalue motion toward the left half plane, i.e., stabilizing, and higher frequencies.

4.4.8 Function Scan Calculations

Function scan calculations begin by setting the system state variable to the current operating point values, and all state variable derivatives to zero.

The system model equations are then evaluated. The specified independent variable, INDEP1, is then set to its initial value, START1, and the model equations are re-evaluated. If the independent variable is a state variable or parameter, the model equations are completely re-evaluated. However, if the independent variable is a variable or rate, which would normally be calculated by the model equations, the re-evaluation beings at the statement immediately following the normal calculation of the variable or rate. In this way, the effect of the variable or rate on the model is determined for the specified, rather than the normal value calculated by the model. This process of re-evaluation is repeated as the independent variable is scanned from START1 to STOP1. After each re-evaluation the value of the specified dependent variable DEPEN is recorded.

If a second independent variable, INDEP2, is specified, this variable is set to its specified value before each scan of INDEP1 and the model is re-evaluated. This places a constraint on the relationship of INDEP1 to INDEP2:

> If INDEP2 is a variable or rate, INDEP1 must be a variable or rate that is calculated below INDEP2 in the model calculation sequence.

If this constraint is violated, INDEP2 will not scan its specified values, but will merely take on its nominal model calculated values. Such a conflict can always be resolved by interchanging INDEP1 and INDEP2. If this form of plots is not desired, the desired family of curves can be obtained by repeated use of the SCAN1 option with INDEP2 varied using the PARAMETER VALUES command.

APPENDIX M

OPTIMAL CONTROLLER DESIGN WITH THE EASY PROGRAM

This appendix is a reproduction of Section 4.5 of reference 1. It presents a description of the optimal controller designs performed by the EASY Analysis Program.

4.5 OPTIMAL CONTROLLER DESIGN

The optimal controller designs performed by the EASY Analysis program are based on the linear optimal regulator theory and linear filter theory of Kalman. By allowing the designer to specify the model order and optimal controller order he wishes to use it is possible to apply the theory to large system models and to obtain reasonable sized practical controller designs.

The design process is shown in Figure 50 where the dashed line indicates engineering feedback needed until the design obtained is acceptable by some criterion. The basic flow indicates the linearization about a desired operating point, the reduction of the linear model, and then the calculation of the optimal gain and filter matrices via linear optimal regulator theory. The initial reduction of order in the linear description is permitted in order to reduce computational and storage requirements in the subsequent controller calculations. Likewise, before leaving the design process, the complexity of the calculated controller can be reduced to any prescribed level to facilitate practical realizations and analysis. The final tasks of preliminary linear analysis (eigenvalues of resultant system with reduced controller) and subsequent simulation of full nonlinear systems with reduced controller are needed to assess the real performance of the design. Based on this, the designing engineer can adjust design parameters to effect more desireable behavior.

Section 4.5.1 considers the model linearization. The method for reduction of the order of linear systems is delayed until Section 4.5.12. The factors affecting the design parameters are considered in Section 4.5.2 where the basic problem definition is given. Section 4.5.3 treats model considerations, including the calculation of default values for design parameters. In Section 4.5.4 the theory for the optimal gain matrix calculation is given. The detailed calculation process is given in

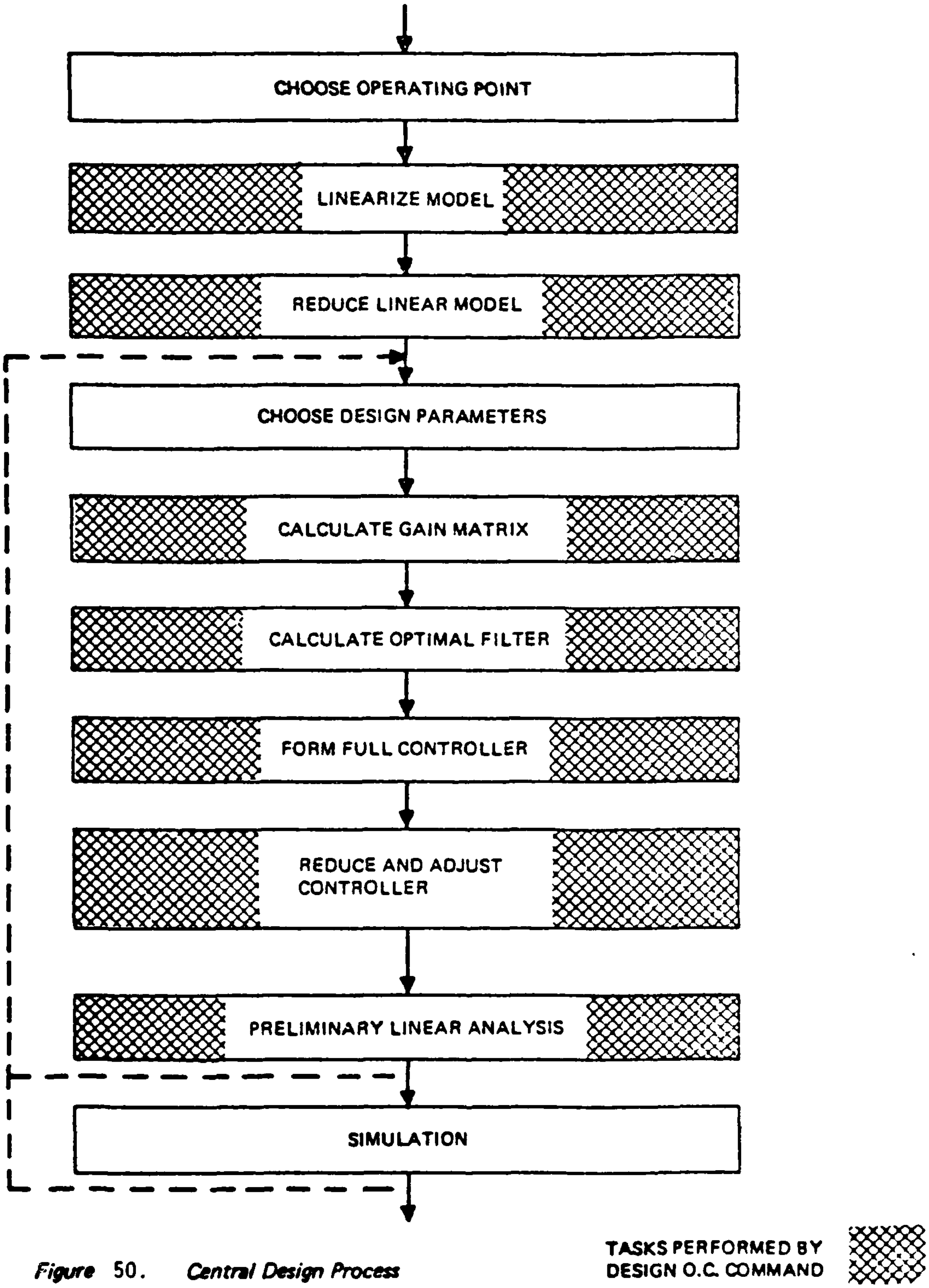


Figure 50. *Central Design Process*

Section 4.5.5 while Section 4.5.6 indicates what analysis information is generated as a result of the calculation process. Section 4.5.7 parallels the development for the Kalman filter with Section 4.5.8 giving the detailed calculation process and Section 4.5.9 the analysis information. Section 4.5.10 then covers the controller formation and subsequent reduction and adjustment. Section 4.5.11 considers the reduction theory with 4.5.12 giving the detailed calculation sequence. Finally, Section 4.5.13 considers the use of the designed controller in the nonlinear system simulation.

4.5.1 Linear Model Generation

The design process starts with the generation of a complete linear model of the system at the specified operating point. This non-linear system model can be expressed by Equations 4.5-1 through 5.4-3.

$$\dot{x} = f(x,u,t) \qquad 4.5\text{-}1$$

$$Y_s = f_x(x,t) \qquad 4.5\text{-}2$$

$$Y_c = f_c(x,u,t) \qquad 4.5\text{-}3$$

where

$\dot{x}$ = n_x dimensional state vector

u = n_u dimensional control vector

Y_s = n_s dimensional sensor vector

Y_c = n_c dimensional criteria vector

f = n_x dimensional vector of nonlinear functions relating state variable, inputs, and time to the state variable derivatives.

f_s = n_s dimensional vector of nonlinear functions relating state variables to sensed quantities.

$f_c =$ n_c dimensional vector of nonlinear functions relating state variables, inputs, and time to criteria quantities.

A linear model of this system is obtained by numerically taking the partial derivatives of f, f_s, and f_c with respect to x and u as described in Section 4.4.3. The equations thus obtained are:

$$\dot{x} = Ax + Bu + I_x d \quad 4.5\text{-}4$$

$$Y_s = H_s x + I_s V \quad 4.5\text{-}5$$

$$Y_c = H_c x + D_c u \quad 4.5\text{-}6$$

where:

$A = n_x$ by n_x system stability matrix

$B = n_x$ by n_u system input matrix

$H_s = n_s$ by n_x system sensor matrix

$H_c = n_c$ by n_x criteria matrix

$D_c = n_c$ by n_u criteria input disturbance matrix

$I_x = n_x$ by n_x identity matrix

$I_s = n_s$ by n_s identity matrix

$d = n_x$ dimensional state disturbance vector

$v = n_s$ dimensional sensor disturbance vector

Note that it is assumed that the control vector, u, of actuator input does not directly effect the sensed quantities, Y_s. The control quantities do effect the sensed quantities via their effect on the system states.

4.5.2 Design Formulation

The state vector x represents deviations from a desired set point and the control vector u represents perturbations about the control level at the set point. The vector d is a disturbance vector for the state derivatives and for this problem is considered to be a zero mean white noise process with a covariance matrix given by a diagonal matrix C_d. Likewise v is a zero mean white noise process affecting the sensors and has a diagonal covariance matrix C_v. With this description, it is to be noted that all set point levels for the state, control, and noise v vectors have been removed. Further, all noise correlation is assumed to be included through additional states representing filtered white noise. Details of this procedure are treated in a later section. The theory presented does not require this limited disturbance description and the design programs can easily be altered to include non-diagonal covariance matrices and a more general multiplier (instead of the identity matrix). The choice was made to facilitate understanding of the design prodedure and to reduce both storage requirements and required input data. Further, the chosen level of generality is sufficient for most all design problems considered in the preliminary design and analyses stages.

The design criterion is given by a cost functional

$$J = \frac{1}{2}\int_0^\infty (y_c' \; Q \, y_c + u' \, Ru) \, dt \qquad 4.5\text{-}7$$

where Q is a positive semi-definite weighting matrix relating the relative importance of the various criteria variables and is assumed diagonal (any off diagonal weighting can be accounted for by a redefinition of the varialbes in the vector y_c). The control weighting matrix R is a positive definite matrix and for convenience assumed diagonal (little physical interpretation can be given to off diagonal terms).

The design problem of interest is to obtain a description of u as a

function of the sensor outputs given by y_s that causes the cost functional of Equation 4.5-7 to be minimized given any initial displacement.

4.5.3 Modeling Considerations

Model Assumptions

Several assumptions are made in the problem description just given for the sake of ease of computing and storage. The zero-mean value assumption for both the state and sensor equations is made knowing that non-zero-mean quantities are included in the set point values. Realizing that equations 4.5-5 through 4.5-6 are for deviations about set point values, the disturbance descriptions are for deviations about their mean values.

The assumption that each state derivative is affected by white noise uncorrelated with that affecting other states seems more restrictive. In practice, however, if one defines band limiting filter equations and accounts for the correlation through the output of the filter entering into the equations for the affected state derivatives, most cases can be approximately treated. The theory that follows does not require this limitation and the computer programs implementing the the algorithms can be modified to include the more general form of the disturbance function. With the limitation, however, the amount of data input and internal storage is reduced.

Design Default Value

From the problem description, the design parameters are the Q and R vectors for the gain calculation and the C_d and D_v vectors for the filter calculation. The defining equations for the criteria variables are also part of the design specification but are more likely to remain fixed for any given problem whereas the Q and R vectors are varied to effect different performing systems. The choice of the elements of Q and R are relative to each other and not absolute (doubling all the

elements of each does not change the problem). Since R must be positive definite a logical default value for any element of R less than or equal to zero is unity. Likewise for Q which must be positive semi-definite, default values are unity for any element less than zero. The above two sets of default values do not take into account any relative sizes of criteria or control variables but only assure the sign definite requirements of the problem formulation.

Default values for the noise covariance matrices (assumed diagonal) used in the calculation of the Kalman Filter require more computation in that they are less likely to be input by design engineers due to less familiarity-especially in the initial stages of the problem. To get some physical interpretation, if one assumes that noise causes errors (both in the state derivatives and in the measurements) that are normally distributed about the correct value with 95% of the errors within a bound $\pm\alpha$, then the appropriate choice for the variance (σ^2) is given by

$$\sigma^2 = \frac{\alpha^2}{3.8416} \qquad 4.5\text{-}8$$

This equation is derived through the use of the erf function as

$$2 \operatorname{erf}\left(\frac{\alpha}{\sigma}\right) = .95 \qquad 4.5\text{-}9$$

or

$$\frac{\alpha}{\sigma} = 1.96 \qquad 4.5\text{-}10$$

which is obtained from a table for the erf function. Equation 4.5-8 is then a direct result of Equation 4.5-10.

To get some bounds on the errors in the calculation of state derivatives due to both external disturbances and model inaccuracy, a measure of the relative size of each state is needed. In the EASY program, this is provided by the ERROR vector. Thus to obtain uncertainty bounds for the

state derivatives, the following equation is used for limit values L^i.

$$L^i = 10 \sum_{j=1}^{n_x} |a_{ij} \cdot \text{ERROR } (j)| \qquad 4.5\text{-}11$$

which indicates the sum of all the absolute state minimum perturbation sizes weighted by the multiplier in the system matrix A. The 10 multiplier is artificial and used to account for model inaccuracy in general and to force the resulting design to favor current measurements rather than historical information (which will happen if the model is assumed more accurate than the measurements) the actual covariance matrix elements is then computed as

$$\sigma_i^2 = L_i^2/3.8416 \qquad 4.5\text{-}12$$

The noise covariance matrix for the measurements is computed in a similar manner where the limits L^i are computed as

$$L^i = \sum_{j-1}^{n_x} |(H_s)_{ij} \cdot \text{ERROR}(j)| \qquad 4.5\text{-}13$$

which weights the measurements relative to the minimum purturbations in the states. This is not ideal but suffices in the absence of any other data.

It is anticipated that these default values will help get a design started but that as experience is gained with the model and with resulting controllers better values can be input to more fully effect the "best" design.

4.5.4 Gain Matrix Calculation

The separation theorem of linear optimal control states that the optimal controller is composed of a linear feedback gain matrix G operating on an optimal estimate of the state obtained through the use of a Kalman filter. The feedback matrix G is computed as if no noise disturbances were present and as if all the states are available for feedback. The following section outlines the procedure for calculating the optimal feeback gain matrix G.

Substitution of the expression for y_c in equation 4.5-6 into the cost functional of equation 4.5-7 yields

$$J = \frac{1}{2} \int_{o}^{\infty} \{H_c x + D_c u)'Q(H_c x + D_c u) + u' Ru\}dt$$

$$= \frac{1}{2} \int_{o}^{\infty} \{x'H_c'QH_c x + u'D_c'QH_c x + x'H_c'QD_c u \qquad 4.5\text{-}14$$

$$+ u' (R+D_c'QD_c)u\}dt$$

Following a procedure using the Minimum Principal of Pontryagin (Ref.2) one forms the Hamiltonian for this system as

$$H = \frac{1}{2} \{x'H_c'QH_c x + u'D_c'QH_c x + x'H_c'QD_c u + u'(R+D_c'QD_c)u\}$$

$$+ p'Ax + p' Bu \qquad 4.5\text{-}15$$

where p is now the costate vector. The differential equation for p is given by

$$\dot{P} = - \frac{\partial H}{\partial x} = - \{H_c'QH_c x + H_c'QD_c u + A'_p\} \qquad 4.5\text{-}16$$

A necessary condition for an optimal solution is given by

$$\frac{\partial H}{\partial u} = 0 = D_c'QH_cx + (R+D_c'QD_c)u + B'p \qquad 4.5\text{-}17$$

which implies

$$u = -\,(R+D_c'QD_c)^{-1}(D_c'QH_cx + B'p). \qquad 4.5\text{-}18$$

Therefore, substitutions of the expression for u into the differential equations for x and p yields

$$\dot{x} = Ax - B\,(R+D_c'QD_c)^{-1}(D_c'QH_cx + B'p) \qquad 4.5\text{-}19$$

$$\dot{p} = -A'p - H_c'QH_cx + H_c'QD_c(R+D_c'QD_c)^{-1}(D_c'QH_cx + B'p) \qquad 4.5\text{-}20$$

or in matrix form

$$\begin{bmatrix} \dot{x} \\ \\ \dot{p} \end{bmatrix} = \left[\begin{array}{c|c} A-B(R+D_c'QD_c)^{-1}D_c'QH_c & -B(R+D_c'QD_c)^{-1}B' \\ & \\ -H_c'(Q-QD_c(R+D_c'QD_c)^{-1}D_c'Q)H_c & -A'+H_c'QD_c(R+D_c'Q)_c^{-1}B' \end{array}\right] \begin{bmatrix} x \\ \\ p \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{A} & -B\tilde{R}^{-1}B' \\ -H_c'\tilde{Q}H_c & -\tilde{A}' \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix} \qquad 4.5\text{-}21$$

where:

$$\tilde{A} = A-B(R+D_c'QD_c)^{-1}D_c'QH_c \qquad 4.5\text{-}22$$

$$\tilde{R} = (R+D_c'QD_c) \qquad 4.5\text{-}23$$

$$\tilde{Q} = Q - Q\ D_c\ (R+D_c'QD_c)^{-1}D_c'Q \qquad 4.5\text{-}24$$

Since R was assumed positive definite and Q positive semi-definite, it can be shown that $\tilde{R}$ is also positive definite and $\tilde{Q}$ is positive semi-definite.

A second condition termed the transversatility condition requires that

$$p(t)\ \Big|_{t\to\infty} = 0. \qquad 4.5\text{-}25$$

When the intial condition for x(t) is considered, it is seen that equations 4.5-21 and 4.5-25 pose a two point boundary value problem. In order to solve for p(t) and x(t) which are needed to determine the control u(t), consider a change of variable

$$\tau = \infty - t \qquad 4.5\text{-}26$$

which when used in equations 4.5-21 and 4.5-25 results in

$$\begin{bmatrix} \dot{x}(\tau) \\ \dot{p}(\tau) \end{bmatrix} = \begin{bmatrix} -\tilde{A} & B\tilde{R}^{-1}B' \\ H_c'\tilde{Q}H_c & \tilde{A}' \end{bmatrix} \begin{bmatrix} x(\tau) \\ p(\tau) \end{bmatrix} \qquad 4.5\text{-}27$$

$$p(\tau)\ \Big|_{\tau=0}=0 \qquad 4.5\text{-}28$$

$$x(\tau)\ _{\tau=\infty}=x. \qquad 4.5\text{-}29$$

Now let Ω be the fundamental* matrix for the system matrix in equation 4.5-27. Partition Ω into quadrants corresponding to the partition in equation 4.5-27 to obtain

$$\begin{bmatrix} x(\tau) \\ p(\tau) \end{bmatrix} = \begin{bmatrix} \Omega_{11}(\tau) & \Omega_{12}(\tau) \\ \Omega_{21}(\tau) & \Omega_{22}(\tau) \end{bmatrix} \begin{bmatrix} x(\tau)|\ \tau = 0 \\ p(\tau)|\ \tau = 0 \end{bmatrix} \qquad 4.5\text{-}30$$

Now using the condition of Equation 4.5-28

$$x(\tau) = \Omega_{11}(\tau) \left[x(\tau)|_{\tau=0} \right] \qquad 4.5\text{-}31$$

$$p(\tau) = \Omega_{21}(\tau) \left[x(\tau)|_{\tau=0} \right] \qquad 4.5\text{-}32$$

from which one obtains

$$p(\tau) = \Omega_{21}(\tau)\Omega_{11}^{-1}(\tau)x(\tau) \qquad 4.5\text{-}33$$

providing $\Omega_{21}(\tau)$ is non singular. Since $\Omega_{11}(\tau)$ is equal to the identify matrix at τ equal to zero and is a fundamental matrix, it is nonsingular for all τ.

Drawing on some results by J. J. O'Donnell, (Ref. 3), it is known that the system matrix of equation 4.5-27 has eigenvalues symmetric with respect to both the real and imaginary axis of the complex plane. This is shown by using a linear transformation

$$J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \qquad 4.5\text{-}34$$

which when applied to the system matrix of equation 4.5-27 indicates it is similar to a matrix whose eigenvalues are the negative of its own.

* Also referred to as the state transition matrix.

The conditions of R and Q being positive definite and semidefinite is sufficient to insure all eigenvalues with zero real parts are of multiplicity 2. Using these facts, let W be a transformation such that

$$W^{-1}\begin{bmatrix} -\tilde{A} & \tilde{B}R^{-1}\tilde{B}' \\ \tilde{H}_c'QH_c & \tilde{A}' \end{bmatrix} W = \begin{bmatrix} \Lambda & 0 \\ 0 & -\Lambda' \end{bmatrix} \tag{4.5-35}$$

where all the eigenvalues of Λ have non-negative real parts and complex eigenvalues occur in conjugate pairs. Thus

$$\Omega(\tau) = W \begin{bmatrix} e^{\Lambda\tau} & 0 \\ 0 & e^{-\Lambda'\tau} \end{bmatrix} W^{-1} \tag{4.5-36}$$

Let

$$U = W^{-1} \tag{4.5-37}$$

and partition U and W to obtain

$$\Omega_{11}(\tau) = W_{11}\, e^{\Lambda\tau}U_{11} + W_{12}\, e^{-\Lambda'\tau}U_{21} \tag{4.5-38}$$

$$\Omega_{21}(\tau) = W_{21}e^{\Lambda\tau}\, U_{11} + W_{22}e^{-\Lambda'\tau}\, U_{21} \tag{4.5-39}$$

Then equation 4.5-33 reduces to

$$p(\tau) = [W_{21}e^{\Lambda\tau}\, U_{11} + W_{22}e^{-\Lambda'\tau}\, U_{21}][W_{11}e^{\Lambda\tau}U_{11} + W_{12}e^{-\Lambda'\tau}\, U_{21}]^{-1}\, x(\tau) \tag{4.5-40}$$

Since we are interested in the control law in the time frame of t near zero, we must look at $p(\tau)$ as τ approaches ∞. If Λ has all eignevalues with positive real parts (not just non-negative) then as τ becomes large the terms with $e^{-\Lambda'\tau}$ must become small with the result that for large τ

$$p(\tau) = \left[W_{21}\, e^{\Lambda\tau} U_{11}\right] \left[W_{11} e^{\Lambda\tau}\, U_{11}\right]^{-1} x(\tau)$$

which assuming non singularity of W_{11} and U_{11} yields

$$p\,(\tau) = W_{21} e^{\Lambda\tau}\, U_{11}\, U_{11}^{-1} (e^{\Lambda\tau})^{-1}\, W_{11}^{-1}\, x(\tau)$$

$$= W_{21}\, W_{11}^{-1}\, x(\tau) \qquad 4.5\text{-}42$$

as τ approaches ∞. Thus for t near zero, from equation 4.5-18 we obtain

$$u(t) = -\,\hat{R}^{-1}\,(D_c{}'QH_c + B'\,W_{21}W_{11}^{-1})\;x(t). \qquad 4.5\text{-}43$$

The condition that causes the indicated inverses W_{11} and U_{11} not to exist is the existence of a unstabilizable mode in the original system equations. If the mode has eigenvalues with zero real parts, the assumption that $e^{-\Lambda'\tau}$ terms in equation 4.5-40 become small with respect to $e^{\Lambda\tau}$ terms is incorrect. If the mode has eigenvalues with positive real parts, then W_{11} will be singular. To see this consider a system of equations

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \end{bmatrix} u \qquad 4.5\text{-}44$$

in which A_1 has eigenvalues with positive real ports.

The resulting matrix for equation 4.5-27 is

$$\begin{bmatrix} \dot{x}_1 (\tau) \\ \dot{x}_2 (\tau) \\ \dot{p}_1 (\tau) \\ \dot{p}_2 (\tau) \end{bmatrix} = \begin{bmatrix} -A_1 & 0 & 0 & 0 \\ 0 & -A_2 & 0 & B_2R^{-1}B_2 \\ Q_{11} & Q_{12} & A_1' & 0 \\ Q_{12} & Q_{22} & 0 & A_2' \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ p_1 \\ p_2 \end{bmatrix} \quad \tau = 0 \qquad 4.5\text{-}45$$

Note now that as one computes the eigenvectors corresponding to eigenvalues with positive real parts the only portion of the eigenvector that can be non-zero is that corresponding to the third partition. Thus W_{11} would have columns of zeros corresponding to each variable in x_1.

The conclusion of this section is that if one is able to partition the eigenvalues as indicated in Equation 4.5-35, and if none of the eigenvalues have zero real parts, and if the inverse of W_{11} exists, then the solution given in 4.5-43 is the correct solution. In practice, the program used to implement the procedure require that the matrix in Equation 4.5-35 be diagonalizable so that if W_{11} is singular, it might also be the result of this restriction not being satisfied.

4.5.5 Solution Process

The numerical process for computing the gain matrix is given by:

1. Form the matrix for the system and adjoint equations as in Equation 4.5-27 with definitions 4.5-22, 4.5-23 and 4.5-24.
2. Compute the eigenvalues of the matrix formed. If any eigenvalues have zero (with the precision of the computation) real parts, this indicates that the system is unstabilizable and that no solution exists. (See Appendix A).

3. Partition the eigenvalues into two groups with all eigenvalues with positive real parts in the first group.
4. Compute eigenvectors for each eigenvalue with a positive real part. (See Appendix A).
5. Partition the eigenvectors computed into matrices W_{11} and W_{21}.
6. Solve for $B'W_{21}W_{11}^{-1}$ where W_{11}^{-1} exists. If W_{11} is singular (within precision limitations) indicate that either the original system had an unstabilizable (unstable and uncontrollable mode or that the rare event of a non-diagonalizable system + adjoint matrix occurred.
7. Compute the gain matrix

$$G = -\tilde{R}^{-1} \left(D'_c Q H_c + B' W_{21} W_{11}^{-1}\right) \qquad 4.5\text{-}46$$

4.5.6 Closed Loop Eigenvalues

Computing the optimal feedback matrix in this manner yields information on the resulting closed loop linear control system. From equation 4.5-35

$$-\tilde{A}\, W_{11} + B\tilde{R}^{-1}\, B' W_{21} = W_{11}\, \Lambda \qquad 4.5\text{-}47$$

Where Λ contained the eigenvalues with positive real parts.

Postmultiplying by $-W_{11}^{-1}$ one obtains

$$\tilde{A} - B\tilde{R}^{-1}\, B'\, W_{21}\, W_{11}^{-1} = W_{11}\, \Lambda\, W_{11}^{-1} \qquad 4.5\text{-}48$$

or when $\tilde{A}$ and $\tilde{R}$ are substituted as in Equation 4.5-22 and 4.5-23

$$A - B(R + D_c' Q D_c)^{-1}(D_c' Q H_c + B' W_{21} W_{11}^{-1}) = W_{11}(-\Lambda) W_{11}^{-1} \qquad 4.5\text{-}49$$

Recognizing the second term as B times the optimal gain matrix G computed in Equation 4.5-46, one obtains

$$A + BG = W_{11} \, (-\Lambda) \, W_{11}^{-1} \qquad 4.5\text{-}50$$

which indicates that the optimal closed loop system given by A+BG has the eigenvalues of $-\Lambda$. For $-\Lambda$ in a diagonal form W_{11} is the set of eigenvectors. Note that as Λ was chosen as all the eigenvalues with positive real parts, $-\Lambda$ must have all eigenvalues with negative real parts. Thus A+BG must be stable.

4.5.7 Kalman Filter Calculation

In this section the filter portion of the total controller is considered. Using the notation of Section 4.5.1 and the results of Theorem 7.1 in the book by Meditch, (Ref. 4), the optimal filtered estimate for the system described in Equations 4.5.4 through 4.5.6 is given by

$$\dot{\hat{x}}\,(t) = A\hat{x}(t) + S(t) \quad Y_s(t) - H_s \, \hat{x}(t) \quad + B \, u(t) \qquad 4.5\text{-}51$$

where

$$\hat{x}(0) = 0$$

and where

$$S(t) = P(t) \, H_s{}' C_v^{-1} \qquad 4.5\text{-}53$$

and where P(t) satisfies the differential equation

$$\dot{P}(t) = A \, P(t) + P(t) \, A' - P(t) \, H_s{}' \, C_v{}^{1} \, H_s \, P(t) + C_d \qquad 4.5\text{-}54$$

with

$$P(0) = E\left[x(0) \, x'(0)\right] \qquad 4.5\text{-}55$$

where the term on the right side of Equation 4.5-55 is the covariance of the state at time zero. Although $P(t)$ and thus $S(t)$ are in general time varying, it is undesireable from an implementation point of view to design time variable controllers. More realistically if one assumes that the covariance of the filtered estimate is at steady state which is obtained as the limiting value of $P(t)$ as t becomes large in Equation 4.5-54, then S given in Equation 4.5-53 becomes a constant matrix with the result that the filter equations are linear and time-invariant.

In order to solve

$$A\,p + P\,A' - PH_s'\,C_v^{-1}\,H_sP + C_d = 0 \qquad 4.5\text{-}56$$

one can use the eigenvector approach reported by Potter (Reference 5) and by O'Donnell (Reference 3) which states that

$$P = W_{21}W_{11}^{-1} \qquad 4.5\text{-}57$$

$$\begin{bmatrix} -A' & H_s'\,C_v^{-1}\,H_s \\ C_d & A \end{bmatrix} \begin{bmatrix} W_{11} \\ W_{21} \end{bmatrix} = \begin{bmatrix} W_{11} \\ W_{21} \end{bmatrix} \Lambda \qquad 4.5\text{-}58$$

and where Λ is the set of eigenvalues of the matrix on the left hand side of equation 4.5-58 that have positive real parts. Then W_{11} and W_{21} are partitions of the set of eigenvectors corresponding to eigenvalues with positive real parts. The solution is analagous to that computed for the gain matrix in the optimal regulator problem and the conditions that all unobservable modes are stable along with C_v positive definite and C_d positive semi-definite insure the existence of W_{11}^{-1} and a solution.

Having calculated W_{21} and W_{11}, the S matrix defined in Equation 4.5-53 can be evaluated from the expression

$$S = W_{21} \; W_{11}^{-1} \, H_s' \, C_v^{-1} \qquad 4.5\text{-}59$$

by first solving the linear system of equations for $W_{11}^{-1} H_s' C_v^{-1}$ and then premultiplying by W_{21}. The dynamic equations for the Kalman filter can now be written (from equation 4.5-51) noting that u(t) is to be given by

$$u = G \hat{x} \qquad 4.5\text{-}60$$

and

$$\dot{\hat{x}} = (A+BG-SH_s)\hat{x} + S \, Y_s \qquad 4.5\text{-}61$$

Equations 4.5-60 and 4.5-61 now form the description of the full controller with x_s the input and u the output.

4.5.8 Kalman Filter Solution Process

er

The numerical process for computing the filter matrix S is given by:

1. Form the $2n_x$ by $2n_x$ matrix of system and adjoined equations given by the left hand side of Equation 4.5-58.
2. Compute the eigenvalues of the matrix formed. If any of the eigenvalues have zero real parts, this is an indication that the system is unobservable and that no solution exists. (See Appendix A for computational details).
3. Partition the eigenvalues into two groups with all the eigenvalues with positive real parts in the first group.
4. Compute the eigenvectors (or real combinations of eigenvectors in the case of complex conjugate eigenvalues) for each eigenvalue in the first group. (See Appendix A for computational details).
5. Partition the matrix computed into W_{11} and W_{21}.
6. Solve for $W_{11}^{-1} \, H_s' \, C_v^{-1}$ with a standard linear equation solver routine. Should W_{11} be singular (or badly conditioned), this indicates that either the original system had an unstable un-

observable mode or that the rare event that the matrix formed in step 1 was undiagonalizable occurred. With the calculation process used, multiple eigenvalues with independent eigenvectors will not cause the method to fail except in extremely rare cases.

7. Compute S as the product of W_{21} with the above solution.

4.5.9 System Eigenvalues Using Kalman Filter

As in the case of the gain matrix calculation where the eigenvalues (obtained by partitioning) with negative real parts were the optimal closed loop eigenvalues for the system using the computed feedback matrix, the eigenvalues computed in the solution process for the Kalman filter have significance.

Using Equations 4.5-60 and 4.5-61 as the description of the full Kalman filter/controller and the original system equations given in 4.5-4 and 4.5-5, one obtains the equations for the total closed loop system as

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & BG \\ SH_s & A+BG-SH_s \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \qquad 4.5\text{-}62$$

Consider now a transformation J where

$$J = \begin{bmatrix} I & 0 \\ I & I \end{bmatrix} \qquad 4.5\text{-}63$$

and where the I's are identity matrices of order n_x.

Then

$$J^{-1}\begin{bmatrix} A & BG \\ SA_s & A+BG - SH_s \end{bmatrix} J$$

$$= \begin{bmatrix} A+BG & BG \\ 0 & A - SH_s \end{bmatrix} \qquad 4.5\text{-}64$$

which indicates that the total closed loop system has eigenvalues corresponding to (A+BG) which are the eigenvalues computed during the calculation of the optimal gain matrix and corresponding to (A-SH_s). It will now be shown that these eigenvalues are the ones computed during the calculation of the Kalman filter. From Equation 4.5-58

$$-A' W_{11} + H_s' C_v^{-1} H_s W_{21} = W_{11} \Lambda \qquad 4.5\text{-}65$$

which postmultiplying by W_{11}^{-1} yields

$$A' - H_s' C_v^{-1} H_s W_{21} W_{11}^{-1} = W_{11} (-\Lambda) W_{11}^{-1} \qquad 4.5\text{-}66$$

Since P from Equation 4.5-56 is symmetric and equal to $W_{21} W_{11}^{-1}$ the use of Equation 4.5-59 yields

$$A' - H_s' S' = W_{11} (-\Lambda) W_{11}^{-1} \qquad 4.5\text{-}67$$

which indicates that the negative of the eigenvalues calculated in the solution process for the S matrix are indeed the eigenvalues of A-SH_s since eigenvalues are invariant under transformation. Thus the $2n_x$ eigenvalues of the total closed loop system are the eigenvalues calculated as part of the gain matrix and optimal filter solution process.

By more manipulation the eigenvectors for the system described in Equation 4.5-62 can be described in terms of the W_{11} matrices (and inverse) calculated for both the gain matrix and Kalman filter. No attempt is made to exploit this information as the real subsequent analysis hinges on a reduced controller operating with the non-linear system.

4.5.10 Controller Formation, Adjustment, and Reduction

The formation of the controller is straightforward when no initial system reduction took place. That is, from equations 4.5-60 and 4.5-61, the controller input is Y_s, the output is the actuator signal u, and the representative block diagram given in Figure 51.

Now the above controller is of the same order (n_x) as the original system description. Since this controller is now just another linear dynamic system, it is natural to ask if a lower order approximation can be made. The input Y_s and output u would have to remain the same but the dynamics describing $\hat{x}$ would be reduced. Section 4.5.11 gives the theory and calculation necessary to reduce this system. For now it suffices to state that a new reduced system of the form shown in Figure 52 results.

Note that in Figure 52 the input and output have not changed. The matrices S_K, G_K, A_K, are now of reduced dimensions (z is not as large as $\hat{x}$) and a new block represented by F_K is added. This is a controller feedforward block and represents a direct gain from the measurements (inputs to the controller) to the control signal (the output from the controller). Intuitively this addition is needed in that when fast dynamics are ignored, their effect is essentially an instantaneous response to the input. Also, the classical methods in control design allow a feedforward controller (i.e. a simple feedback gain) so that this reduction process that results in the F_K term seems most reasonable.

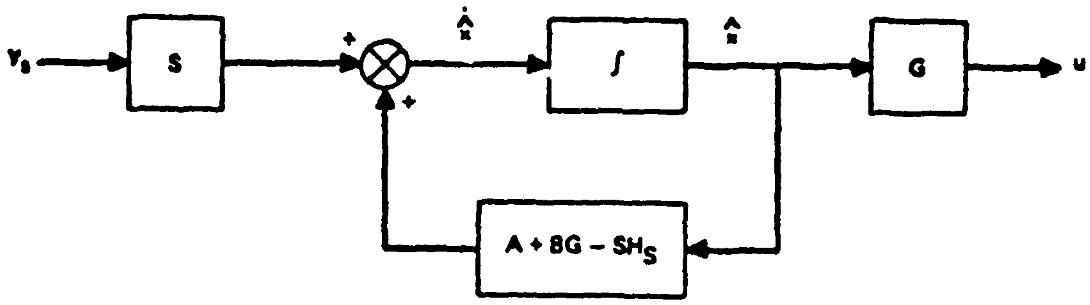


Figure 51. *Full Controller Block Diagram*

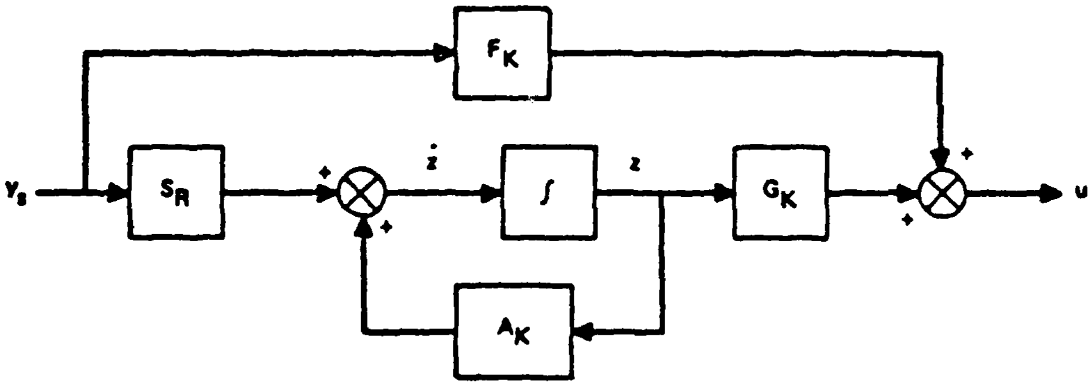


Figure 52. *Reduced Controller Block Diagram*

Controller formation and reduction for the case when initial system reduction took place is more complicated. If one proceeds in a logical manner for the reduction of order in the initial description, a feedforward term in the expression for the sensor output Y_s results, that is

$$Y_s = H_s x + D_s u \qquad 4.5\text{-}68$$

In block diagram form, the initial system (reduced) appears in Figure 53.

This term due to D_s does not have any effect during the calculation of the optimal gain matrix and can be ignored during the calculation of the optimal filter. That is, the optimal filter is predicted on an input $H_s x$ which is now really $(Y_s - D_s u)$. Thus to form a controller with input Y_s and output u, one has to subtract the $D_s u$ term so that the correct input to the Kalman filter results. This is shown in Figure 54.

The total controller is now the dynamics between points P1 and P2. Several alternatives now exist for the reduction of this controller. Since the feedback term involving D_s has no dynamics associated, order reduction can be accomplished either before or after simplification by elimination of the feedback path. Elimination before results in a system shown in Figure 55.

This system is now just like the one shown in Figure 51 except for the extra term in the system matrix and can be similarily reduced. Another approach would be to take the dynamic system between points P2 and P3 in Figure 54 which is now just that of Figure 51 and reduce it to obtain the system shown in Figure 52. If this is done, and the reduced system substituted between points P2 and P3 in Figure 56, the following block diagram results.

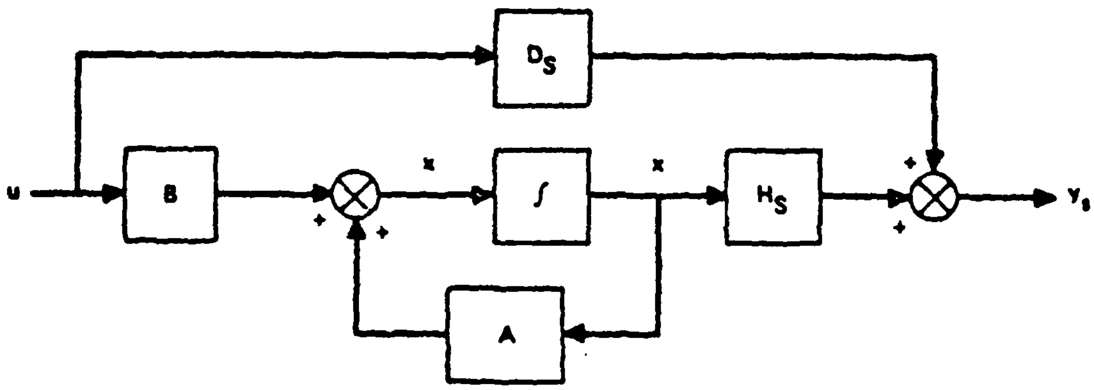


Figure 53. *Reduced System Block Diagram*

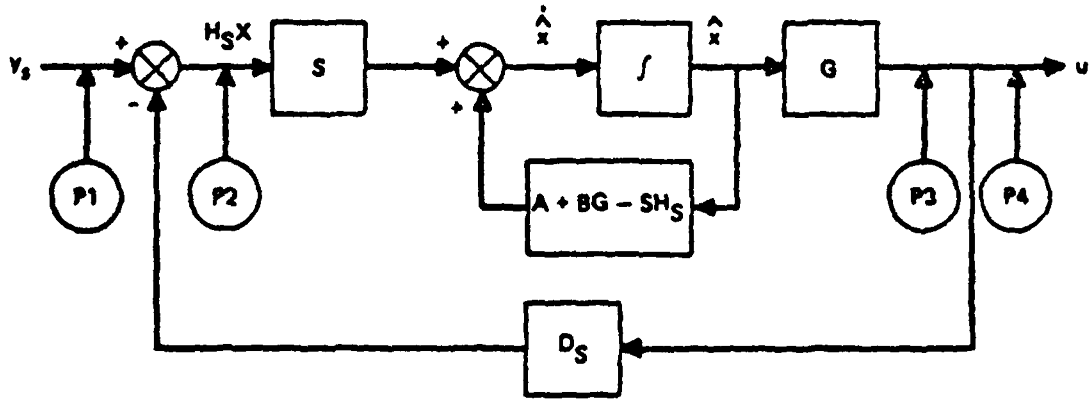


Figure 54. *Block Diagram of Controller When Initial System Had Feedforward Term*

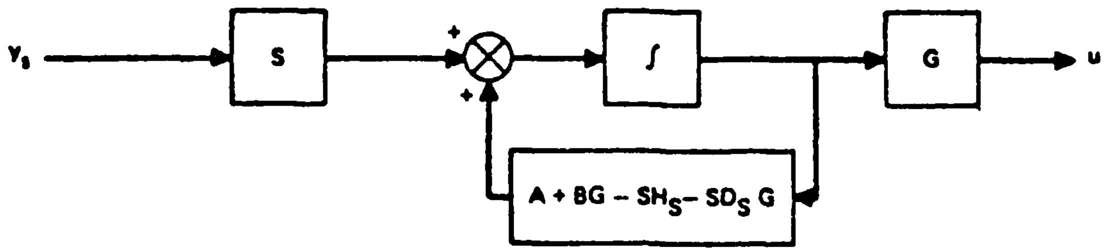


Figure 55. *Total Controller Incorporating Static Feedback*

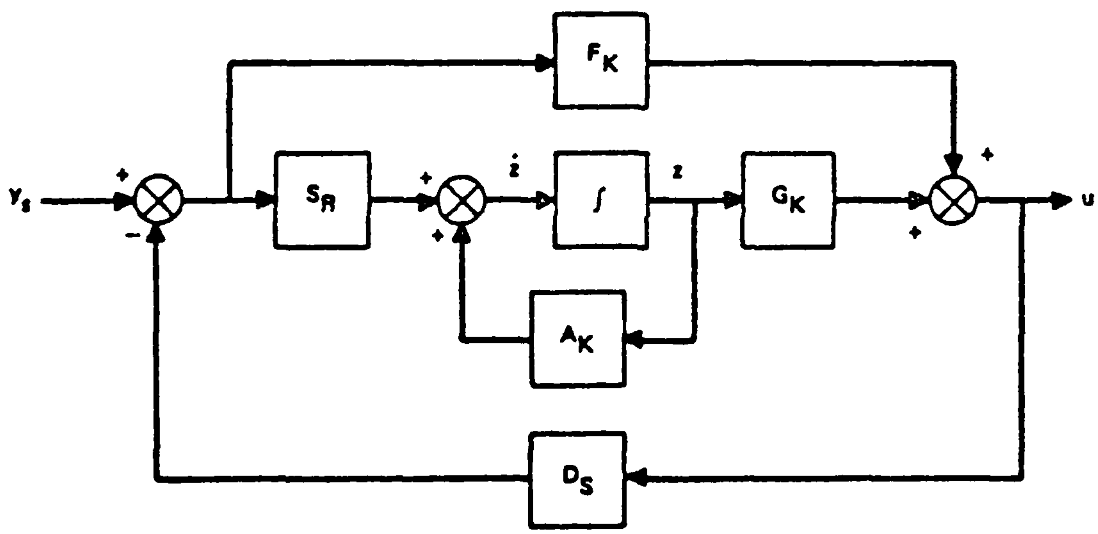


Figure 56. *Reduced Controller Before Static Feedback Elimination*

In the block diagram of Figure 56 there is both algebraic feed-forward and feedback that must be accounted for. Noting that for the case shown in Figure 56.

$$u = F_K\,(Y_s - D_s\,u) + G_K z \qquad 4.5\text{-}69$$

Then one can solve for u as

$$u = (I + F_K D_S)^{-1}(F_K Y_S + G_K z) \qquad 4.5\text{-}70$$

providing the inverse exists (an assumption rarely violated). From this one can define a modified F_K and G_K as

$$\bar{F}_K \equiv (I + F_K D_S)^{-1} F_K \qquad 4.5\text{-}71$$

$$\bar{G} \equiv (I + F_K D_S)^{-1} G_K \qquad 4.5\text{-}72$$

with

$$u = \bar{F}_K\, Y_S + \bar{G}_K\, z. \qquad 4.5\text{-}73$$

The expression for the dynamic portion then becomes

$$\dot{z} = A_K\, z + S_R\,(Y_S - D_S u) \qquad 4.5\text{-}74$$

which through the use of Equation 4.5-73 becomes

$$\begin{aligned} \dot{z} &= A_K z + S_R\, Y_S - S_R D_S\, \bar{F}_K\, Y_S - S_R D_S\, \bar{G}_K \\ &= (A_K - S_R D_S \bar{G}_K) z + (S_R - S_R D_S \bar{F}_K)\, Y_S \end{aligned} \qquad 4.5\text{-}75$$

which indicate the modified A_K and S_K required to eliminate the static feedback. By using this second technique, the linear analysis of the resulting system becomes simple as

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A + B F_K H_S & BG_K \\ S_K H_S & A_K \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \quad 4.5\text{-}76$$

For this reason and because little advantage is seen for either technique over the other, the second method is the one implemented.

4.5.11 Linear System Reduction Theory

The problem of interest is to approximate a high order linear dynamic system by one of lower order in such a manner that the output responses due to various inputs are "close". The value of a low order approximation lies in the reduced computational and storage requirements for analysis and design and in the reduced complexity for implementation. Consider the linear description in the normal form,

$$\dot{x} = Ax + Bu \quad 4.5\text{-}77$$

$$y = Hx \quad 4.5\text{-}78$$

Where X is a n_x dimensional state vector, u is a n_u dimensional control vector and Y is a n_s dimensional measurement vector. The lower order approximation sought is of the form

$$\dot{z} = A_R z + B_R u \quad 4.5\text{-}79$$

$$y = H_R z + D_R u \quad 4.5\text{-}80$$

where u and y are as defined above and z is a n_R dimensional reduced state vector with

$$n_R \leq n_x \quad 4.5\text{-}81$$

This description differs from many reported in the literature in that the feedforward term accounted for in D_R is permitted. In some cases this may be a disadvantage but for most, especially for the simplification of controllers, it leads to a natural and appealing reduction.

The proposed approach is a classical one of retaining the lowest frequency modes and neglecting the dynamics associated with the higher stable modes.

Consider a transformation T where T is nonsingular and

$$x = T\,w \qquad 4.5\text{-}82$$

and

$$T^{-1} A\, T = \Lambda = \begin{bmatrix} \Lambda_L & 0 \\ 0 & \Lambda_H \end{bmatrix} \qquad 4.5\text{-}83$$

Where Λ is block diagonal with 1 by 1 blocks for real eigenvalues and 2 by 2 blocks for complex conjugate pairs of eigenvalues. For this discussion and for the implementation, it is assumed that A is diagonalizable (any multiple eigenvalues have as many independent eigenvectors). Further it is assumed that Λ is partitioned into Λ_L and Λ_H where all unstable and the lowest magnitude stable eigenvalues are in Λ_L and the large magnitude stable eigenvalues and in Λ_H. The resulting equations for a similarily partitioned w are

$$\begin{bmatrix} \dot{w}_L \\ \dot{w}_H \end{bmatrix} = \begin{bmatrix} \Lambda_L & 0 \\ 0 & \Lambda_H \end{bmatrix} \begin{bmatrix} w_L \\ w_H \end{bmatrix} + \begin{bmatrix} (T^{-1}\,B)_L \\ (T^{-1}\,B)_H \end{bmatrix} u \qquad 4.5\text{-}84$$

$$y = \begin{bmatrix} (H\;T)_L & (H\;T)_H \end{bmatrix} \begin{bmatrix} w_L \\ w_H \end{bmatrix} \qquad 4.5\text{-}85$$

To neglect the dynamics associated with w_H is to assume that w_H responds instantaneously to any input. Thus $\dot{w}_H$ should be zero resulting in

$$\dot{w}_H = \Lambda_H w_H + (T^{-1} B)_H u = 0 \qquad 4.5\text{-}86$$

$$w_H = -\Lambda_H^{-1} (T^{-1} B)_H u \qquad 4.5\text{-}87$$

which is the algebraic relation desired. Equations 4.5-84 and 4.5-85 can then be written eliminating w_H as

$$\dot{w}_L = \Lambda_L w_L + (T^{-1} B)_L u \qquad 4.5\text{-}88$$

$$y = (H T)_L w_L - (H T)_H \Lambda_H^{-1} (T^{-1} B)_H u \qquad 4.5\text{-}89$$

with the terms identified as

$$A_R = \Lambda_L \qquad 4.5\text{-}90$$

$$B_R = (T^{-1} B)_L \qquad 4.5\text{-}91$$

$$H_R = (H Y)_L \qquad 4.5\text{-}92$$

$$D_R = -(H T)_H \Lambda_H^{-1} (T^{-1} B)_H \qquad 4.5\text{-}93$$

The needed assumption is that Λ_H^{-1} exists which will be the case when Λ_H contains large stable eigenvalues. Note also that n_R can be prespecified as long as n_R is greater than the number of unstable eigenvalues. Further, it may be necessary to adjust n_R one integer less to insure that Λ_L is partitioned such that both of complex conjugate eigenvalues are included or excluded.

For this reduction technique, the reduced model is asymptotically correct for any input level. As the eigenvalues in Λ_H become separated from those in Λ_L, the approximation naturally becomes more exact.

4.5.12 Reduction Calculation Sequence

The numerical process for computing the reduced linear system consists of:

1. Compute the eigenvalues of the full A matrix.
2. Sort the eigenvalues according to real parts with most positive at the top. Count unstable eigenvalues to insure retention.
3. Compute eigenvectors for sorted list.
4. Compute T^{-1} B and H T and partition.
5. Set A_R as block diagonal matrix mode from computed eigenvalues at top of list.
6. Set B_R and H_R as the top partitions in T^{-1} B and H T respectively.
7. Compute D_R as $-(H\ T)_H \Lambda_H^{-1} (T^{-1}\ B)_H$

4.5.13 Controller Use In Simulation

The controller designed is returned to the simulation program as a linear system described by the four matrices F_K, A_K, G_K, and S_K. It must be remembered, however, that all the design analysis was performed about an operating point defined by u_o and y_o. For a total controller, these quantities must be added back in. A total controller block diagram is thus given in Figure 57.

If several controllers are designed around several operating points, it may be necessary to "gain schedule" by changing controllers and set points as a function of operating point measured or commanded. These and other decisions on the value of the designed controller must now be based on the results of the simulation.

4.6 WARNING MESSAGES

One or more of the following warning messages will occur if the program encounters difficulty in interpreting analysis instructions or performing an analysis. These messages will be preceded by: ***WARNING***.

The symbols xxx, zzz, or nnn are used to indicate phrases from the analysis description that are included as part of the warning message. The following messages are listed in alphabetical order:

AD-A096 597 BOEING MILITARY AIRPLANE CO SEATTLE WA F/G 1/3
ANALYSIS OF EJECTION SEAT STABILITY USING EASY PROGRAM. VOLUME --ETC(U)
SEP 80 C L WEST, B R UMMEL, R F YURCZYK F33615-79-C-3407

UNCLASSIFIED AFWAL-TR-80-3014-VOL-1 NL

8 OF 8

AD A
096597

END
DATE
FILMED
4-81
DTIC

APPENDIX N

EASIEST EXAMPLE

This appendix presents a supplementary ejection seat analysis example. This example utilizes the following components which were not included in the ejection seat simulation example in Section VI.

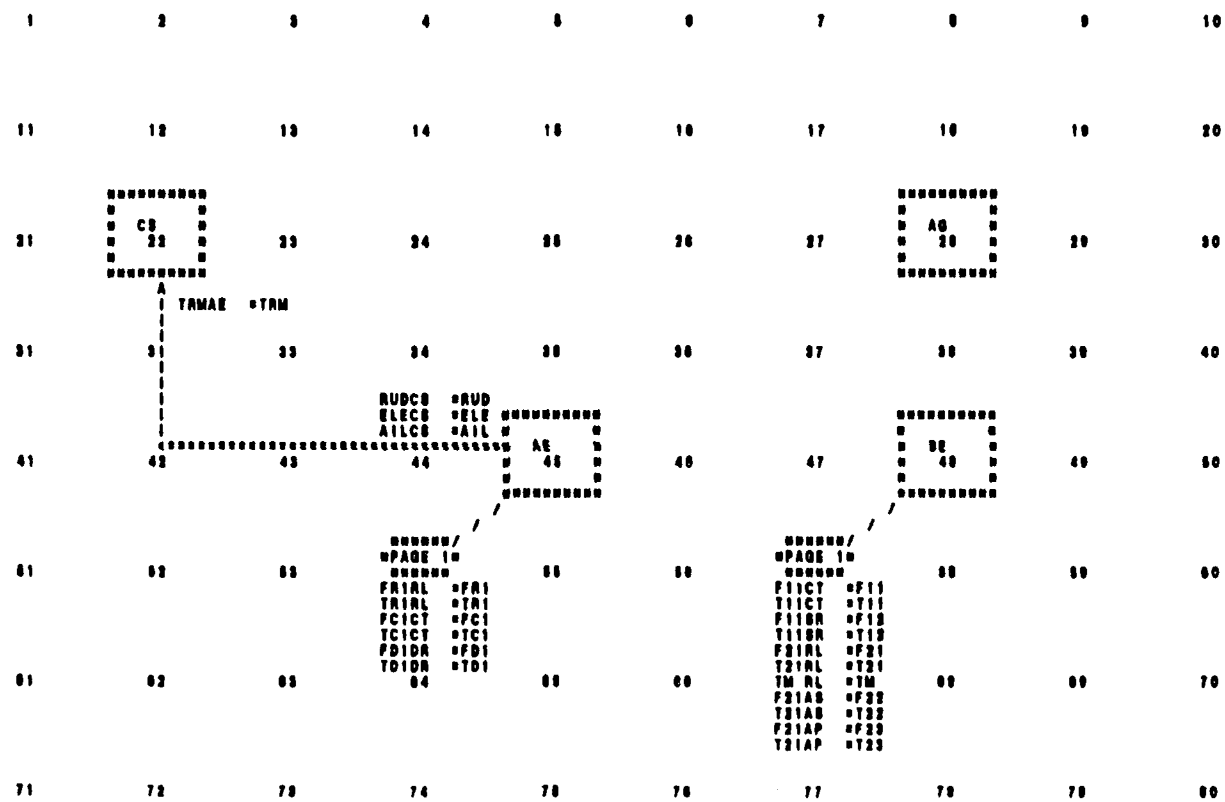
o	AE	Airplane
o	CS	Airplane control surfaces
o	DR	Dart
o	AP	Aerodynamic plate

A simplified thrust vector control system is also included in this model.

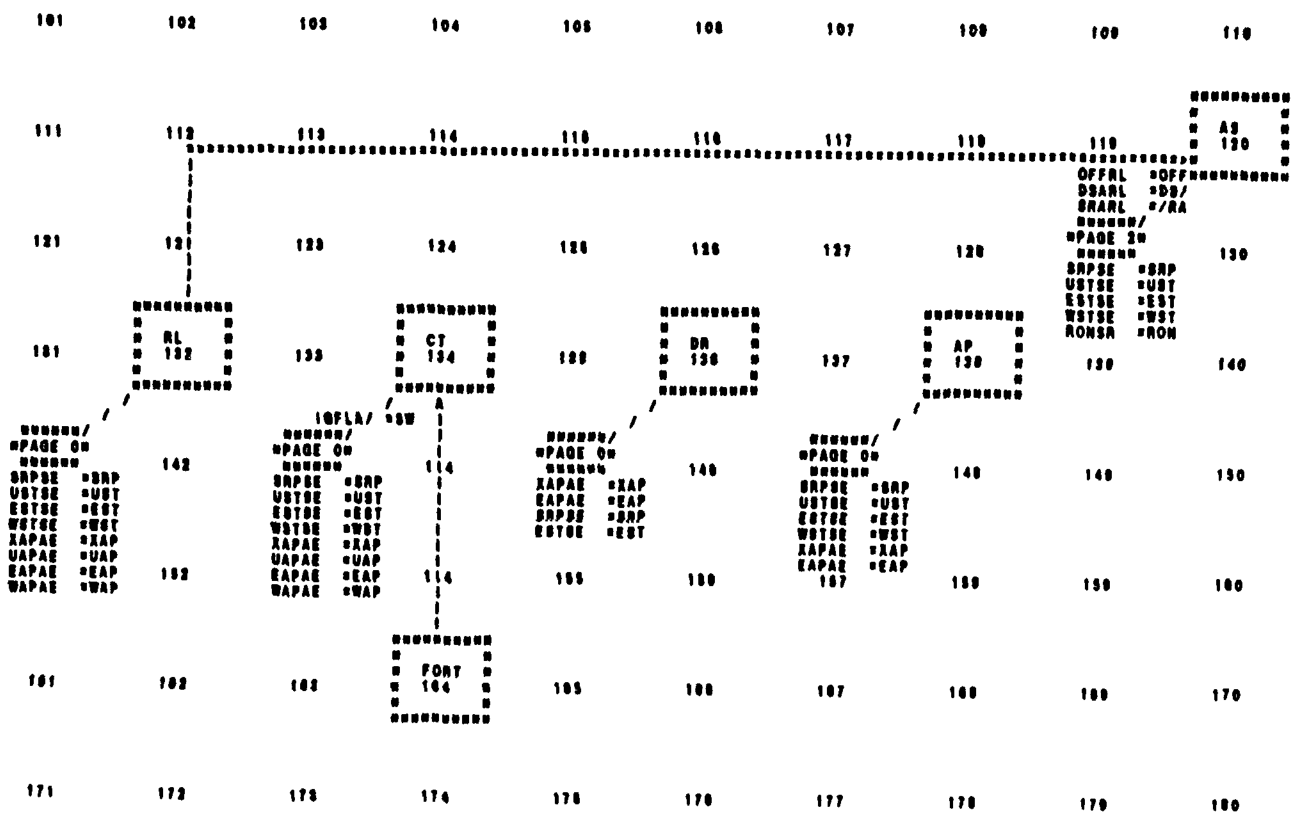
```
MODEL DESC = EASIEST EXAMPLE WITH FLUIDIC THRUST VECTOR CONTROL
LOCATION=020   AG
*
LOCATION=104   FORT
                                  ADD VARIABLES=IGFLAG
                                  ADD PARAMETERS=CTTIME
FORTRAN STATEMENTS
        IGFLAG=0
        IF(TIME.GE.CTTIME)IGFLAG=1
*
LOCATION=134   CT        INPUTS=SE(SRP=SRP,UST=UST,EST=EST,WST=WST)
                                AE(XAP=XAP,UAP=UAP,EAP=EAP,WAP=WAP)
                                FORT(IGFLAG=SW)
LOCATION=250   SR        INPUTS=IT2(I8,2=PIT),CT
LOCATION=130   DR        INPUTS=SE(SRP=SRP,EST=EST),AE(XAP=XAP,EAP=EAP)
LOCATION=130   AP        INPUTS=SE(SRP=SRP,UST=UST,EST=EST,WST=WST)
                                AE(XAP=XAP,EAP=EAP)
LOCATION=132   RL        INPUTS=SE(SRP=SRP,UST=UST,EST=EST,WST=WST)
                                AE(XAP=XAP,UAP=UAP,EAP=EAP,WAP=WAP)
LOCATION=022   CS        INPUTS=AE
LOCATION=045   AE        INPUTS=RL(1=1),CT(1=1),DR(1=1)
LOCATION=120   AS        INPUTS=SE(SRP=SRP,UST=UST,EST=EST,WST=WST)
                                RL,SR(RON=RON)
LOCATION=040   SE        INPUTS=CT(1=1),SR(1=2),RL(1=1),AS(1=2)
                                AP(1=3)
*
LOCATION=233   FORT      INPUTS=RONSR,WSTSE,S2 IN1
                             ADD VARIABLES=G1,X1DOT,B1OUT
                             ADD PARAMETERS=B1
FORTRAN STATEMENTS
C
C   *****  RATES AS MEASURED BY THE FTVC HARDWARE  *****
C
C
C   B1      = POLE ASSOCIATED WITH PSEUDO INTEGRATOR
C   G1      = PSEUDC INTEGRATOR DC GAIN
C   X1DOT   = PSEUDO INTEGRATOR RATE (DEG/SEC)
C   S2 IN1  = PSEUDO INTEGRATOR STATE (DEG)
C   WSTSE   = SEAT PITCH RATE (DEG/SEC)
C
        IF(INST.NE.20)
       .G1 = (-ALOG((B1+SQRT(400.+B1**2))/20.)+ALOG(B1+SQRT(1.+B1**2)))/
       .       (ARTAN2(20.,B1)-ARTAN2(1.,B1))/B1
        X1DOT = (-B1*S2 IN1 + G1*B1*WSTSE(2)) * RONSR
        B1OUT = B1
*
LOCATION=215   IN1       INPUTS=FORT(X1DOT=8,1)
*
LOCATION=237   FORT      INPUTS=G1,B1OUT,RONSR,S2 IN1,S2 IT2,WSTSE
                             ADD VARIABLES=X2DOT,G2,G3
                             ADD PARAMETERS=A1,B2,GDC,X1BIAS,RKINIT
FORTRAN STATEMENTS
C
C   *****  SUSTAINER ROCKET NOZZLE DEFLECTION  *****
C
C   A1      = ZERO ASSOCIATED WITH PSEUDO INTEGRATOR
C   B2      = POLE ASSOCIATED WITH PSEUDO INTEGRATOR
C   G2      = RATE GAIN
C   G3      = ATTITUDE GAIN
C   GDC     = CONTROL SYSTEM DC GAIN
C   X2DOT   = SUSTAINER ROCKET DEFLECTION RATE (DEG/SEC)
C   S2 IT2  = SUSTAINER ROCKET DEFLECTION (DEG)
C   X1BIAS  = PITCH BIAS ANGLE (DEG)
```

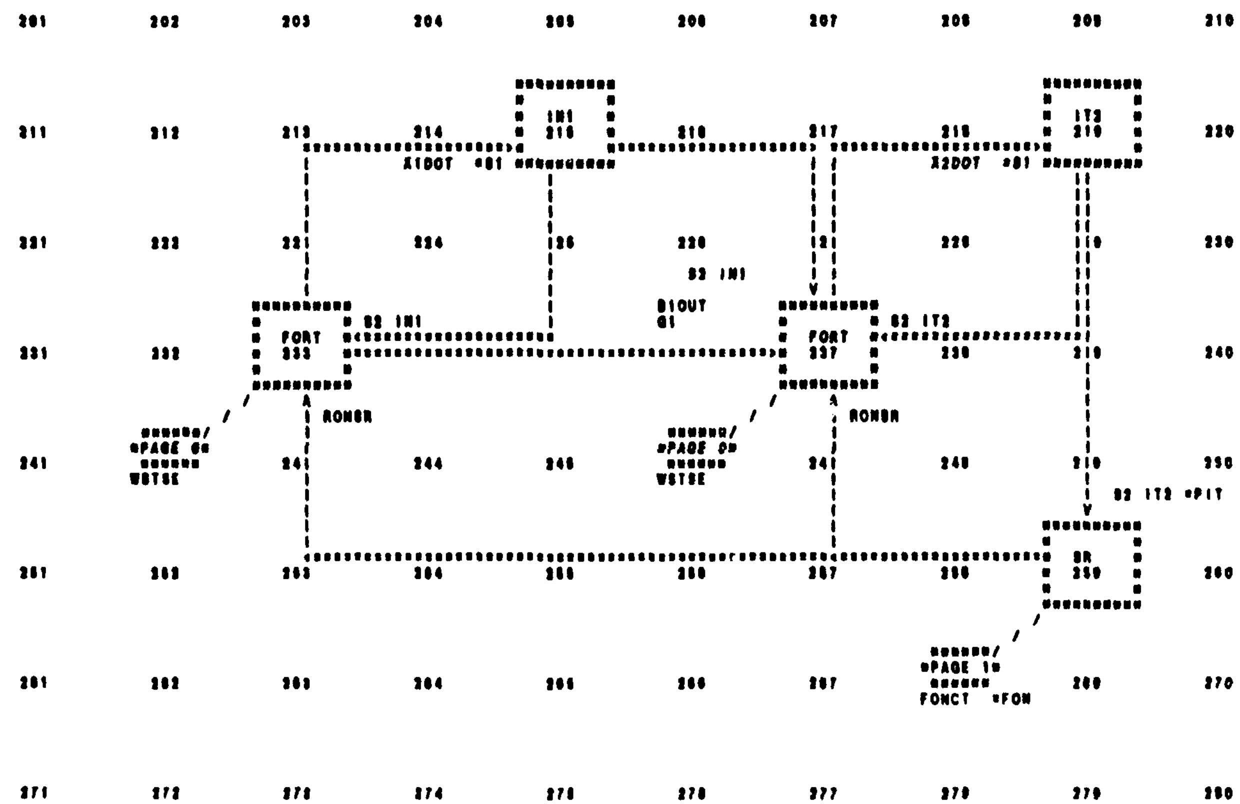
```
C
      IF((NST.EQ.26) GO TO 140
      Q2=(B1OUT*QDC)/A1
      Q3=(QDC-Q2)/Q1
 140  X2DOT = -B2*(S2 (T2-RK(NIT-Q2*WSTSE(2))-Q3*(S2 (N1-X1B(AS))*RONSR
*
LOCATION=210   IT2      INPUTS=FORT(X2DOT=S,1)
END OF MODEL
PRINT
```

EASIEST EXAMPLE WITH FLUIDIC THRUST VECTOR CONTROL PAGE 0



EASIEST EXAMPLE WITH FLUIDIC THRUST VECTOR CONTROL




INI
215
IT2
219
X1DOT *01
X2DOT *01
S2 INI
S1OUT
G1
S2 IT2
FORT
233
FORT
237
RONSR
PAGE 6
WSTSE
SR
259
S2 IT2 *PIT
PAGE 1
FONCT *FON

```
*  ------  EASIEST ANALYSIS FILE EXAMPLE  ------
*
*
*  **********************************************
*  ****************  TABLES  ********************
*  **********************************************
*
*
*  -----  TABLE FOR COMPONENT "CT" (CATAPULT)  -----
*
TABLE      TCPCT,10
*  CATAPULT PROPELLANT WEB CONSUMED (INCHES)
0.0        6.310E-3   1.502E-2   3.080E-2   4.880E-2
7.630E-2   1.081E-1   1.172E-1   1.284E-1   1.422E-1
*  CATAPULT PROPELLANT CONSUMED (SLUGS)
0.0        1.110E-4   1.860E-4   4.010E-4   6.240E-4
8.770E-4   1.231E-3   1.347E-3   1.472E-3   1.703E-3
*
*  -----  TABLE FOR COMPONENT "SR" (SUSTAINER ROCKET)  -----
*
TABLE      TRFSR,7
*  TIME INTO SUSTAINER ROCKET BURN (SEC)
0.0        .001       .030       .140       .280       .310       .350
*  SUSTAINER ROCKET FORCE (LBS)
0.0        3425.      2830.      3400.      3150.      1000.      0.00
*
*  -----  TABLE FOR COMPONENT "DR" (DART)  -----
*
TABLE      TBFDR,2
*  LINE LENGTH (FT)
0.0        10.0
*  LINE FORCE (LB)
1100.      1100.
*
*  -----  TABLES FOR COMPONENT "AP" (ATTACHED PLATE)  -----
*
TABLE      TCXAP,9
*  PLATE ANGLE OF ATTACK (DEG)
-90.       -60.       -45.       -30.       0.      30.     45.     60.     90.
*  X-AXIS FORCE COEFFICIENT
 0.        -0.33      -0.35      -0.17      0.      -0.17   -0.35   -0.33   0.
*
TABLE      TCZAP,9
*  PLATE ANGLE OF ATTACK (DEG)
-90.       -60.       -45.       -30.       0.      30.     45.     60.     90.
*  Z-AXIS FORCE COEFFICIENT
 1.0       0.60       0.35       0.20       0.      -0.20   -0.35   -0.60   -1.0
*
*  -----  TABLE FOR COMPONENT "AS" (SEAT AERO)  -----
*
TABLE      TAEAS,9
*  CREWMAN EXPOSED LENGTH (FT)
0.00    0.25    0.50    0.75    1.00    4.25    4.75    5.25    5.50
*  CREWMAN EXPOSED AREA (FT**2)
0.00    0.15    0.40    0.62    0.84    5.20    5.53    5.85    5.90
*
*********************************************************
********************  PARAMETER VALUES  *****************
*********************************************************
*
*  -----  PARAMETERS FOR COMPONENT "AG" (ATMOSPHERE AND GRAVITY)  -----
*
PARAMETER VALUES
SW AG=0         WINAG=2
*
```

```
*  -----  PARAMETERS FOR COMPONENT "CT" (CATAPULT)  -----
*
CTTIME=.001
SAPCT=-.88,0,-3.2                        AAPCT=13.81,0,-0.330
UCLCT=4.2208      CSKCT=2.833            VI CT=10.8            PA CT=0.788
PT CT=10          CSPCT=8000             C  CT=.001703         CI CT=.00001
PMWCT=25.8        SK CT=100000           CK CT=800             GAMCT=1.199
TF CT=2600        C1 CT=.120             C2 CT=0.3             B  CT=.0333
SXPCT=.44         T1 CT=204              UP CT=1               TDECT=.08
*
*  -----  PARAMETERS FOR COMPONENT "SR" (SUSTAINER ROCKET)  -----
*
RKINIT=-53.78
PCGSR=-.81,0,-1.88                       EA SR=0,4.83,0
XRNSR=0,0,1.60
YAWSR=0           PL SR=2.817            PODSR=0.2187          PIDSR=0.1488
*
*  -----  PARAMETERS FOR COMPONENT "DR" (DART)  -----
*
DAPDR=14.08,0,-.338         DBADR=0.11,0,2.65
*
*  -----  PARAMETERS FOR COMPONENT "AP" (ATTACHED PLATE)  -----
*
XPCAP=-2,0,-1.8   PA AP=10   EPLAP=0,-18,0   ZEMAP=-8.80
*
*  -----  PARAMETERS FOR COMPONENT "RL" (RAILS)  -----
*
BL1RL =  -0.4128    0.0870    0.0333
BL2RL =  -0.4713    0.0870   -0.1140
BL3RL =  -0.5011    0.0870   -1.2501
BL4RL =  -0.4128   -0.0870    0.0333
BL5RL =  -0.4713   -0.0870   -0.1140
BL6RL =  -0.5011   -0.0870   -1.2501
RLRRL=-4.041      XRRRL=13.81, 0.087,-0.330
RLLRL=-4.041      XRLRL=13.81,-0.087,-0.330
ERLRL=0,17,0
SPRRL=75000,75000          DPGRL=200,200
SDFRL=0.02                 ZTSRL=-3.0
*
*  -----  PARAMETERS FOR COMPONENT "CS" (CONTROL SURFACES)  -----
*
COECS=-18.        TDECS=.3               TCECS=1
*
*  -----  PARAMETERS FOR COMPONENT "AE" (AIRPLANE)  -----
*
S  AE=530         C  AE=10.04            AW AE=43282.          XCPAE=0
B  AE=38.888      AMIAE=20202,135940,170839
APIAE=4828,0,0    XENAE=3.344,0,-.308    ENDAE=.988,0,-.001
TALAE=900         TVEAE=800
*
*  -----  PARAMETERS FOR COMPONENT "AS" (SEAT AERODYNAMICS)  -----
*
ZWSAS=-8.80       XEMAS=0,0,-3.25                              CDXAS=0
ECXAS=1           ECYAS=1                ECZAS=1               CLPAS=-4.838E-4
CMOAS=-1.047E-2                          CNRAS=-8.238E-4
S  AS=0.84
*
*  ----  PARAMETERS FOR COMPONENT "SE" (SEAT)  ----
*
CW SE=402    CCGSE= 455,.018,-.042   CMISE=18.72,31.14,8.27
CPISE= 62,3.78,-.05
*
*  -----  PARAMETERS FOR THE THRUST VECTOR CONTROL  -----
*
A1=10       B1=0.4      B2=120      QDC=1.25      XIBIAS=12
```

```
GKIINI=1   GKIIT2=1   GKLIT2=800   AMAIT2=-30   AMIIT2=-80
*
**** INITIALIZATION
*
INITIAL CONDITIONS=W1 SR=5.8,S2 INI=0,S2 IT2=0
                   XAPAE=0,0,-900    UAPAE=700,0,42
                   EAPAE=0,3,0       WAPAE=0,0,0
                   USTSE=777,0,191   SRPSE=14,0,-801.7
                   ESTSE=0,14,0
CALC XIC
INT CONTROLS   EAPAE(1)=0,XAPAE=0,0,1
               AILCS=0,ELECS=0,RUDCS=0
               EF CT=0,EL CT=0,WK CT=0,WS CT=0
               W1 SR=0,S2 INI=0,S2 IT2=0
SS ITERATIONS=100
STEADY STATE
PARAMETER VALUES
SW AG=1
XIC-X
STEADY STATE
*
***  SIMULATION
*
ALL STATES
XIC-X
PRINT CONTROL=8
PRINT VARIABLES
SRPSE(1),SRPSE(2),SRPSE(3)
ESTSE(1),ESTSE(2),ESTSE(3)
XAPAE(1),XAPAE(2),XAPAE(3)
EAPAE(1),EAPAE(2),EAPAE(3)
F21AP(1),F21AP(2),F21AP(3)
T21AP(1),T21AP(2),T21AP(3)
F21DR(1),F21DR(2),F21DR(3)
T21DR(1),T21DR(2),T21DR(3)
ALPAS,BETAS,VM AS
ELECS,S2 IT2
PRINTER PLOTS
TITLE=EASIEST EXAMPLE
DISPLAY1,USTSE(1),VS,TIME
         USTSE(2),VS,TIME
         USTSE(3),VS,TIME
         SRPSE(1),VS,TIME
         SRPSE(2),VS,TIME
DISPLAY2,SRPSE(3),VS,TIME
         WSTSE(1),VS,TIME
         WSTSE(2),VS,TIME
         WSTSE(3),VS,TIME
         ESTSE(1),VS,TIME
DISPLAY3,ESTSE(2),VS,TIME
         ESTSE(3),VS,TIME
         F21AP(1),VS,TIME
         F21AP(2),VS,TIME
         F21AP(3),VS,TIME
DISPLAY4,T21DR(1),VS,TIME
         T21DR(2),VS,TIME
         T21DR(3),VS,TIME
         ALPAS,VS,TIME
         BETAS,VS,TIME
DISPLAY5,S2 IT2,VS,TIME
INT MODE=2,TMAX=0.80,TINC=.001
PRATE=2,OUTRATE=10
SIMULATE
```

```
                              /*/*/*/   SIMULATION ANALYSIS   /*/*/*/

PRATE =   2   OUTRATE =  10   PRINT CONTROL =   6   INT MODE =   2   TINC =   .10000E-02   TMAX =   .80000
PRATE2 =   2   OUTRATE2 =  10   PRINTS =   5   PRINT2 FROM  .10000E+37 TO  .10000E+37   TINC2 =   .10000E-02

                                          EASIEST EXAMPLE

           CASE NO.   3                         80/12/11.   13.28.15.

         TIME = 0.
SRPSE  (1)    =  13.920     SRPSE  (2)    = -.10000E-05 SRPSE  (3)    = -901.72     ESTSE  (1)    =  .48083E-03
ESTSE  (2)    =  13.781     ESTSE  (3)    =  .14303E-02 XAPAE  (1)    = 0.          XAPAE  (2)    = 0.
XAPAE  (3)    = -900.00     EAPAE  (1)    = 0.          EAPAE  (2)    =  1.3003     EAPAE  (3)    =  .10300E-20
F21AP  (1)    = 0.          F21AP  (2)    = 0.          F21AP  (3)    = 0.          T21AP  (1)    = 0.
T21AP  (2)    = 0.          T21AP  (3)    = 0.          F21DR  (1)    = 0.          F21DR  (2)    = 0.
F21DR  (3)    = 0.          T21DR  (1)    = 0.          T21DR  (2)    = 0.          T21DR  (3)    = 0.
ALPAS         = 0.          BETAS         = 0.          VM AS         = 0.          ELECS         = 0.
S2 IT2        = 0.

    CATAPULT IGNITION AT TIME =      .0014 SEC

         TIME =  .2000E-01
SRPSE  (1)    =  28.920     SRPSE  (2)    =  .93840E-08 SRPSE  (3)    = -901.72     ESTSE  (1)    =  .84438E-03
ESTSE  (2)    =  13.738     ESTSE  (3)    =  .18338E-02 XAPAE  (1)    =  16.000     XAPAE  (2)    =  .20443E-05
XAPAE  (3)    = -900.00     EAPAE  (1)    = -.87888E-07 EAPAE  (2)    =  1.3003     EAPAE  (3)    = -.05380E-05
F21AP  (1)    = 0.          F21AP  (2)    = 0.          F21AP  (3)    = 0.          T21AP  (1)    = 0.
T21AP  (2)    = 0.          T21AP  (3)    = 0.          F21DR  (1)    = 0.          F21DR  (2)    = 0.
F21DR  (3)    = 0.          T21DR  (1)    = 0.          T21DR  (2)    = 0.          T21DR  (3)    = 0.
ALPAS         = 0.          BETAS         = 0.          VM AS         = 0.          ELECS         = 0.
S2 IT2        = -30.000

         TIME =  .4000E-01
SRPSE  (1)    =  48.027     SRPSE  (2)    = -.82732E-08 SRPSE  (3)    = -901.73     ESTSE  (1)    =  .37898E-03
ESTSE  (2)    =  13.705     ESTSE  (3)    =  .20851E-02 XAPAE  (1)    =  32.000     XAPAE  (2)    =  .58807E-05
XAPAE  (3)    = -900.00     EAPAE  (1)    = -.26808E-08 EAPAE  (2)    =  1.3080     EAPAE  (3)    = -.55270E-04
F21AP  (1)    = 0.          F21AP  (2)    = 0.          F21AP  (3)    = 0.          T21AP  (1)    = 0.
T21AP  (2)    = 0.          T21AP  (3)    = 0.          F21DR  (1)    = 0.          F21DR  (2)    = 0.
F21DR  (3)    = 0.          T21DR  (1)    = 0.          T21DR  (2)    = 0.          T21DR  (3)    = 0.
ALPAS         = 0.          BETAS         = 0.          VM AS         = 0.          ELECS         = 0.
S2 IT2        = -30.000

         TIME =  .6000E-01
SRPSE  (1)    =  61.920     SRPSE  (2)    = -.88810E-08 SRPSE  (3)    = -901.78     ESTSE  (1)    =  .84850E-03
ESTSE  (2)    =  13.350     ESTSE  (3)    =  .38264E-02 XAPAE  (1)    =  48.000     XAPAE  (2)    =  .88807E-05
XAPAE  (3)    = -900.00     EAPAE  (1)    = -.11278E-08 EAPAE  (2)    =  1.3077     EAPAE  (3)    = -.25333E-03
F21AP  (1)    = 0.          F21AP  (2)    = 0.          F21AP  (3)    = 0.          T21AP  (1)    = 0.
T21AP  (2)    = 0.          T21AP  (3)    = 0.          F21DR  (1)    = 0.          F21DR  (2)    = 0.
F21DR  (3)    = 0.          T21DR  (1)    = 0.          T21DR  (2)    = 0.          T21DR  (3)    = 0.
ALPAS         = 0.          BETAS         = 0.          VM AS         = 0.          ELECS         = 0.
S2 IT2        = -30.000

         TIME =  .8000E-01
SRPSE  (1)    =  77.000     SRPSE  (2)    = -.40220E-04 SRPSE  (3)    = -901.81     ESTSE  (1)    = -.57392E-03
ESTSE  (2)    =  13.332     ESTSE  (3)    =  .78793E-02 XAPAE  (1)    =  64.000     XAPAE  (2)    =  .11303E-04
XAPAE  (3)    = -900.00     EAPAE  (1)    = -.28208E-08 EAPAE  (2)    =  1.3030     EAPAE  (3)    = -.82174E-03
F21AP  (1)    = 0.          F21AP  (2)    = 0.          F21AP  (3)    = 0.          T21AP  (1)    = 0.
T21AP  (2)    = 0.          T21AP  (3)    = 0.          F21DR  (1)    = 0.          F21DR  (2)    = 0.
F21DR  (3)    = 0.          T21DR  (1)    = 0.          T21DR  (2)    = 0           T21DR  (3)    = 0.
ALPAS         = 0.          BETAS         = 0.          VM AS         = 0.          ELECS         = 0.
S2 IT2        = -30.000

         TIME =  .1000E+00
```

```
END OF GUIDED STROKE AT TIME =   .2180 SEC

          TIME =  .2200
SRPSE  (1)     =  180.01      SRPSE  (2)     = -.18714E-02 SRPSE  (3)     = -804.30      ESTSE  (1)     = -.13181E-01
ESTSE  (2)     =  13.387      ESTSE  (3)     =  .76312E-02 XAPAE  (1)     =  176.01      XAPAE  (2)     =  .27076E-04
XAPAE  (3)     = -890.87      EAPAE  (1)     =  .43813E-04 EAPAE  (2)     =  1.1880      EAPAE  (3)     = -.22144E-01
F21AP  (1)     = 0.           F21AP  (2)     = 0.          F21AP  (3)     = 0.           T21AP  (1)     = 0.
T21AP  (2)     = 0.           T21AP  (3)     = 0.          F21DR  (1)     = 0.           F21DR  (2)     = 0.
F21DR  (3)     = 0.           T21DR  (1)     = 0.          T21DR  (2)     = 0.           T21DR  (3)     = 0.
ALPAS          =  10.867      BETAS          =  .13018E-01 VM AS          =  .70878      ELECS          = 0.
S2 IT2         = -42.738

    AERODYNAMIC PLATE PENETRATION AT TIME =      .2340 SEC

    SEAT/RAIL SEPARATION AT TIME   .2370 SEC

          TIME =  .2400
SRPSE  (1)     =  204.78      SRPSE  (2)     = -.18608E-02 SRPSE  (3)     = -808.21      ESTSE  (1)     = -.32811E-01
ESTSE  (2)     =  13.647      ESTSE  (3)     =  .42434E-01 XAPAE  (1)     =  182.01      XAPAE  (2)     =  .14838E-04
XAPAE  (3)     = -890.86      EAPAE  (1)     =  .11784E-04 EAPAE  (2)     =  1.0883      EAPAE  (3)     = -.26310E-01
F21AP  (1)     = -238.77      F21AP  (2)     = 0.          F21AP  (3)     =  233.66      T21AP  (1)     = 0.
T21AP  (2)     =  822.48      T21AP  (3)     = 0.          F21DR  (1)     = 0.           F21DR  (2)     = 0.
F21DR  (3)     = 0.           T21DR  (1)     = 0.          T21DR  (2)     = 0.           T21DR  (3)     = 0.
ALPAS          =  10.884      BETAS          =  .40707E-01 VM AS          =  .70778      ELECS          = 0.
S2 IT2         = -67.386

          TIME =  .2600
SRPSE  (1)     =  220.47      SRPSE  (2)     = -.10748E-02 SRPSE  (3)     = -805.88      ESTSE  (1)     = -.10378
ESTSE  (2)     =  12.643      ESTSE  (3)     =  .80058E-01 XAPAE  (1)     =  208.01      XAPAE  (2)     = -.39861E-05
XAPAE  (3)     = -890.84      EAPAE  (1)     = -.32103E-04 EAPAE  (2)     =  1.0818      EAPAE  (3)     = -.30173E-01
F21AP  (1)     = -317.85      F21AP  (2)     = 0.          F21AP  (3)     =  313.67      T21AP  (1)     = 0.
T21AP  (2)     =  1104.1      T21AP  (3)     = 0.          F21DR  (1)     = 0.           F21DR  (2)     = 0.
F21DR  (3)     = 0.           T21DR  (1)     = 0.          T21DR  (2)     = 0.           T21DR  (3)     = 0.
ALPAS          =  8.8400      BETAS          =  .11744     VM AS          =  .70381      ELECS          = 0.
S2 IT2         = -82.485

          TIME =  .2800
SRPSE  (1)     =  238.08      SRPSE  (2)     =  .20041E-03 SRPSE  (3)     = -806.78      ESTSE  (1)     = -.20488
ESTSE  (2)     =  10.781      ESTSE  (3)     =  .88301E-01 XAPAE  (1)     =  224.02      XAPAE  (2)     = -.27424E-04
XAPAE  (3)     = -890.83      EAPAE  (1)     = -.78281E-04 EAPAE  (2)     =  1.0266      EAPAE  (3)     = -.33424E-01
F21AP  (1)     = -418.73      F21AP  (2)     = 0.          F21AP  (3)     =  410.27      T21AP  (1)     = 0.
T21AP  (2)     =  1444.1      T21AP  (3)     = 0.          F21DR  (1)     = 0.           F21DR  (2)     = 0.
F21DR  (3)     = 0.           T21DR  (1)     = 0.          T21DR  (2)     = 0.           T21DR  (3)     = 0.
ALPAS          =  7.8788      BETAS          =  .27782     VM AS          =  .70081      ELECS          = 0.
S2 IT2         = -84.812

          TIME =  .3000
SRPSE  (1)     =  251.64      SRPSE  (2)     =  .17288E-02 SRPSE  (3)     = -807.85      ESTSE  (1)     = -.53085
ESTSE  (2)     =  8.1224      ESTSE  (3)     =  .63420E-01 XAPAE  (1)     =  240.02      XAPAE  (2)     = -.58288E-04
XAPAE  (3)     = -890.82      EAPAE  (1)     = -.12688E-03 EAPAE  (2)     =  .89400      EAPAE  (3)     = -.35880E-01
F21AP  (1)     = -487.48      F21AP  (2)     = 0.          F21AP  (3)     =  480.84      T21AP  (1)     = 0.
T21AP  (2)     =  1728.1      T21AP  (3)     = 0.          F21DR  (1)     = 0.           F21DR  (2)     = 0.
F21DR  (3)     = 0.           T21DR  (1)     = 0.          T21DR  (2)     = 0.           T21DR  (3)     = 0.
ALPAS          =  6.3112      BETAS          =  .84278     VM AS          =  .68829      ELECS          = -.10887E-02
S2 IT2         = -85.030

          TIME =  .3200
SRPSE  (1)     =  267.14      SRPSE  (2)     =  .30710E-02 SRPSE  (3)     = -808.30      ESTSE  (1)     = -.82701
ESTSE  (2)     =  7.8078      ESTSE  (3)     =  .88368E-01 XAPAE  (1)     =  256.02      XAPAE  (2)     = -.80887E-04
XAPAE  (3)     = -890.80      EAPAE  (1)     = -.17878E-03 EAPAE  (2)     =  .86824      EAPAE  (3)     = -.37788E-01
F21AP  (1)     = -582.14      F21AP  (2)     = 0.          F21AP  (3)     =  644.88      T21AP  (1)     = 0.
T21AP  (2)     =  1818.0      T21AP  (3)     = 0.          F21DR  (1)     = 0.           F21DR  (2)     = 0.
```

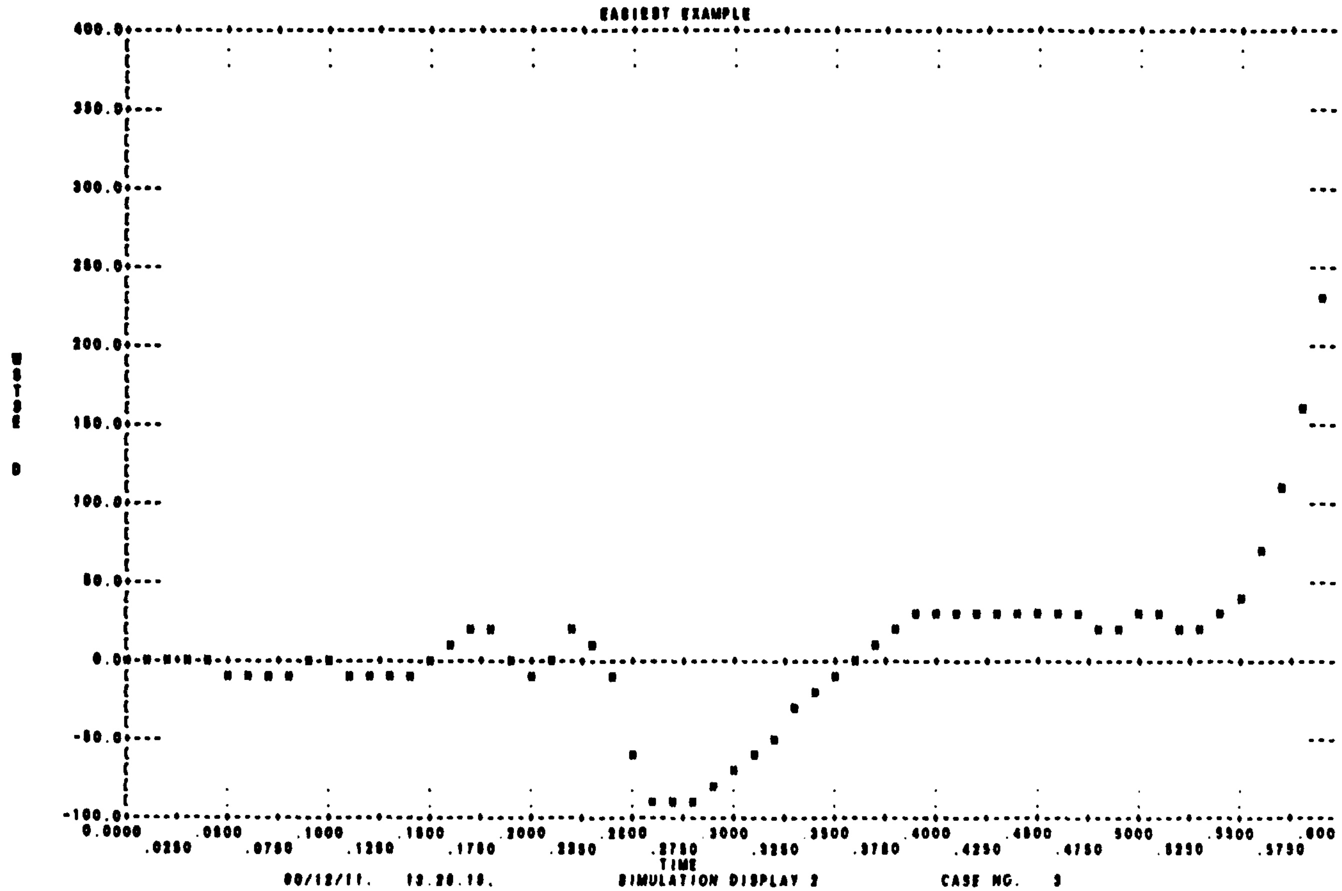
```
     END OF GUIDED STROKE AT TIME =    .2180 SEC

          TIME =  .2200
SRPSE  (1)     =  188.01      SRPSE  (2)     = -.18714E-02 SRPSE  (3)     = -904.80      ESTSE  (1)     = -.13181E-01
ESTSE  (2)     =  13.387      ESTSE  (3)     =  .70312E-02 XAPAE  (1)     =  178.01      XAPAE  (2)     =  .27078E-04
XAPAE  (3)     = -909.87      EAPAE  (1)     =  .43813E-04 EAPAE  (2)     =  1.1388      EAPAE  (3)     = -.22144E-01
F21AP  (1)     = 0.           F21AP  (2)     = 0.          F21AP  (3)     = 0.           T21AP  (1)     = 0.
T21AP  (2)     = 0.           T21AP  (3)     = 0.          F21DR  (1)     = 0.           F21DR  (2)     = 0.
F21DR  (3)     = 0.           T21DR  (1)     = 0.          T21DR  (2)     = 0.           T21DR  (3)     = 0.
ALPAS          =  10.887      BETAS          =  .13018E-01 VM AS          =  .70878      ELECS          = 0.
S2 IT2         = -42.738

     AERODYNAMIC PLATE PENETRATION AT TIME =      .2340 SEC

     SEAT/RAIL SEPARATION AT TIME   .2378 SEC

          TIME =  .2400
SRPSE  (1)     =  204.78      SRPSE  (2)     = -.18008E-02 SRPSE  (3)     = -908.21      ESTSE  (1)     = -.32811E-01
ESTSE  (2)     =  13.047      ESTSE  (3)     =  .42434E-01 XAPAE  (1)     =  182.01      XAPAE  (2)     =  .14535E-04
XAPAE  (3)     = -908.88      EAPAE  (1)     =  .11784E-04 EAPAE  (2)     =  1.0893      EAPAE  (3)     = -.28310E-01
F21AP  (1)     = -238.77      F21AP  (2)     = 0.          F21AP  (3)     =  233.88      T21AP  (1)     = 0.
T21AP  (2)     =  832.48      T21AP  (3)     = 0.          F21DR  (1)     = 0.           F21DR  (2)     = 0.
F21DR  (3)     = 0.           T21DR  (1)     = 0.          T21DR  (2)     = 0.           T21DR  (3)     = 0.
ALPAS          =  10.884      BETAS          =  .40707E-01 VM AS          =  .70778      ELECS          = 0.
S2 IT2         = -67.368

          TIME =  .2600
SRPSE  (1)     =  220.47      SRPSE  (2)     = -.30748E-03 SRPSE  (3)     = -908.88      ESTSE  (1)     = -.10378
ESTSE  (2)     =  12.643      ESTSE  (3)     =  .80058E-01 XAPAE  (1)     =  208.01      XAPAE  (2)     = -.39081E-05
XAPAE  (3)     = -908.84      EAPAE  (1)     = -.33102E-04 EAPAE  (2)     =  1.0818      EAPAE  (3)     = -.30173E-01
F21AP  (1)     = -317.88      F21AP  (2)     = 0.          F21AP  (3)     =  313.87      T21AP  (1)     = 0.
T21AP  (2)     =  1104.1      T21AP  (3)     = 0.          F21DR  (1)     = 0.           F21DR  (2)     = 0.
F21DR  (3)     = 0.           T21DR  (1)     = 0.          T21DR  (2)     = 0.           T21DR  (3)     = 0.
ALPAS          =  9.0480      BETAS          =  .11744     VM AS          =  .70351      ELECS          = 0.
S2 IT2         = -83.455

          TIME =  .2800
SRPSE  (1)     =  238.09      SRPSE  (2)     =  .20041E-03 SRPSE  (3)     = -908.78      ESTSE  (1)     = -.28489
ESTSE  (2)     =  10.761      ESTSE  (3)     =  .88301E-01 XAPAE  (1)     =  224.02      XAPAE  (2)     = -.27424E-04
XAPAE  (3)     = -908.93      EAPAE  (1)     = -.70281E-04 EAPAE  (2)     =  1.0288      EAPAE  (3)     = -.33424E-01
F21AP  (1)     = -418.73      F21AP  (2)     = 0.          F21AP  (3)     =  410.27      T21AP  (1)     = 0.
T21AP  (2)     =  1444.1      T21AP  (3)     = 0.          F21DR  (1)     = 0.           F21DR  (2)     = 0.
F21DR  (3)     = 0.           T21DR  (1)     = 0.          T21DR  (2)     = 0.           T21DR  (3)     = 0.
ALPAS          =  7.6768      BETAS          =  .27782     VM AS          =  .70081      ELECS          = 0.
S2 IT2         = -94.812

          TIME =  .3000
SRPSE  (1)     =  251.64      SRPSE  (2)     =  .17288E-02 SRPSE  (3)     = -907.85      ESTSE  (1)     = -.53086
ESTSE  (2)     =  9.1224      ESTSE  (3)     =  .83420E-01 XAPAE  (1)     =  240.02      XAPAE  (2)     = -.56258E-04
XAPAE  (3)     = -898.82      EAPAE  (1)     = -.13888E-03 EAPAE  (2)     =  .88400      EAPAE  (3)     = -.35880E-01
F21AP  (1)     = -487.46      F21AP  (2)     = 0.          F21AP  (3)     =  480.84      T21AP  (1)     = 0.
T21AP  (2)     =  1728.1      T21AP  (3)     = 0.          F21DR  (1)     = 0.           F21DR  (2)     = 0.
F21DR  (3)     = 0.           T21DR  (1)     = 0.          T21DR  (2)     = 0.           T21DR  (3)     = 0.
ALPAS          =  6.3112      BETAS          =  .84278     VM AS          =  .68528      ELECS          = -.10887E-02
S2 IT2         = -85.030

          TIME =  .3200
SRPSE  (1)     =  267.14      SRPSE  (2)     =  .30710E-02 SRPSE  (3)     = -908.30      ESTSE  (1)     = -.82701
ESTSE  (2)     =  7.8078      ESTSE  (3)     =  .89388E-01 XAPAE  (1)     =  258.02      XAPAE  (2)     = -.80887E-04
XAPAE  (3)     = -898.90      EAPAE  (1)     = -.17875E-03 EAPAE  (2)     =  .86524      EAPAE  (3)     = -.37786E-01
F21AP  (1)     = -552.14      F21AP  (2)     = 0.          F21AP  (3)     =  544.88      T21AP  (1)     = 0.
T21AP  (2)     =  1918.0      T21AP  (3)     = 0           F21DR  (1)     = 0            F21DR  (2)     = 0
```

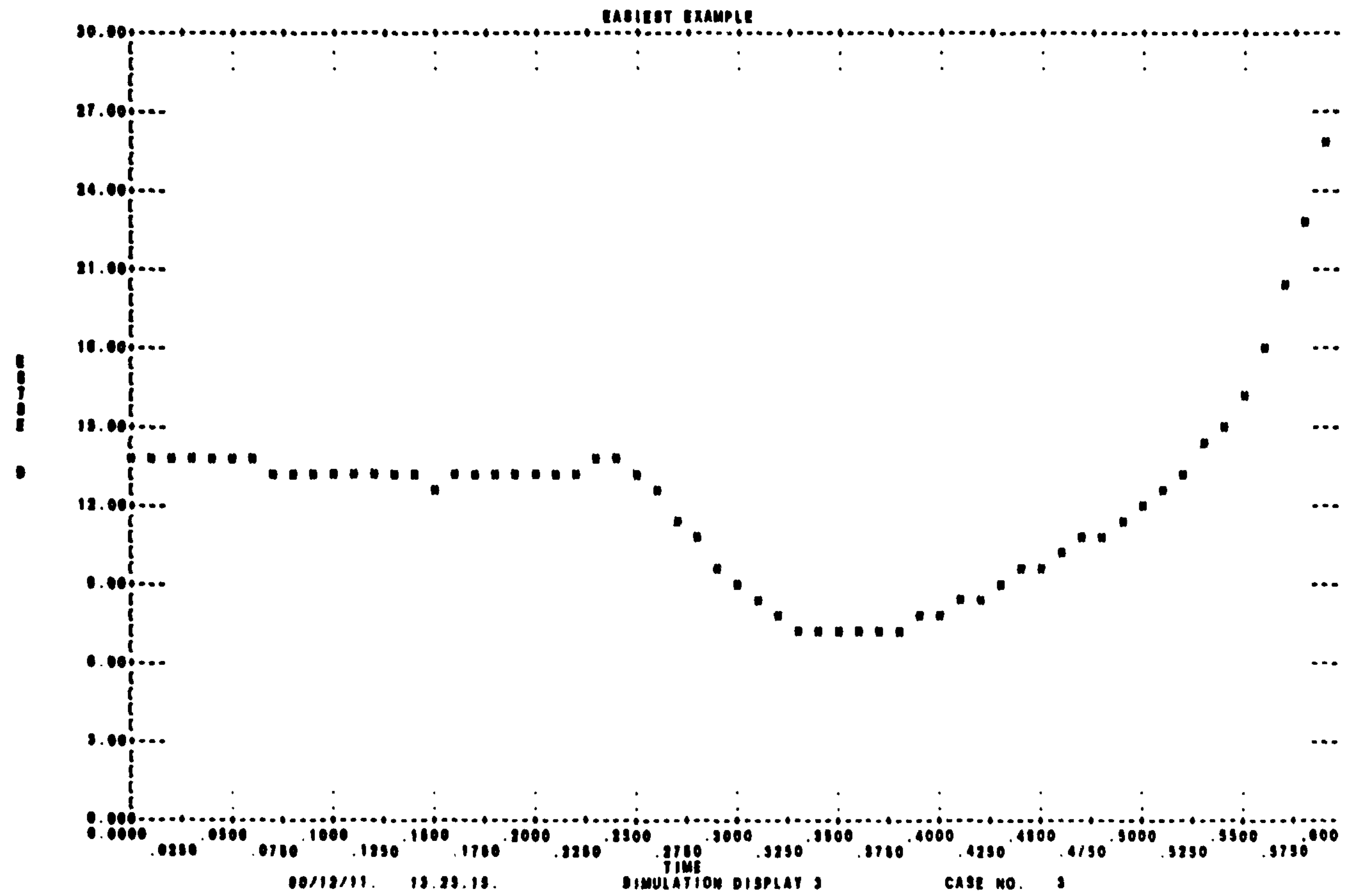
```
SRPSE  (1)    =  93.800     SRPSE  (2)    = -.83314E-04 SRPSE  (3)    = -901.92     ESTSE  (1)    = -.82231E-03
ESTSE  (2)    =  13.227     ESTSE  (3)    =  .01308E-02 XAPAE  (1)    =  90.001     XAPAE  (2)    =  .14285E-04
XAPAE  (3)    = -900.00     EAPAE  (1)    = -.36180E-05 EAPAE  (2)    =  1.2862     EAPAE  (3)    = -.10302E-02
F21AP  (1)    = 0.          F21AP  (2)    = 0.          F21AP  (3)    = 0.          T21AP  (1)    = 0.
T21AP  (2)    = 0.          T21AP  (3)    = 0.          F21DR  (1)    = 0.          F21DR  (2)    = 0.
F21DR  (3)    = 0.          T21DR  (1)    = 0.          T21DR  (2)    = 0.          T21DR  (3)    = 0.
ALPAS         = 0.          BETAS         = 0.          VM AS         = 0.          ELECS         = 0.
S2 IT2        = -30.000

         TIME =  .1200
SRPSE  (1)    =  108.70     SRPSE  (2)    = -.13880E-03 SRPSE  (3)    = -902.13     ESTSE  (1)    = -.40782E-04
ESTSE  (2)    =  13.112     ESTSE  (3)    =  .08803E-02 XAPAE  (1)    =  96.001     XAPAE  (2)    =  .17788E-04
XAPAE  (3)    = -900.00     EAPAE  (1)    = -.18087E-08 EAPAE  (2)    =  1.2834     EAPAE  (3)    = -.37441E-02
F21AP  (1)    = 0.          F21AP  (2)    = 0.          F21AP  (3)    = 0.          T21AP  (1)    = 0.
T21AP  (2)    = 0.          T21AP  (3)    = 0.          F21DR  (1)    = 0.          F21DR  (2)    = 0.
F21DR  (3)    = 0.          T21DR  (1)    = 0.          T21DR  (2)    = 0.          T21DR  (3)    = 0.
ALPAS         = 0.          BETAS         = 0.          VM AS         = 0.          ELECS         = 0.
S2 IT2        = -30.000

         TIME =  .1400
SRPSE  (1)    =  125.70     SRPSE  (2)    = -.27873E-03 SRPSE  (3)    = -902.41     ESTSE  (1)    = -.21718E-02
ESTSE  (2)    =  12.938     ESTSE  (3)    =  .70287E-02 XAPAE  (1)    =  112.00     XAPAE  (2)    =  .21104E-04
XAPAE  (3)    = -900.00     EAPAE  (1)    =  .21463E-08 EAPAE  (2)    =  1.2843     EAPAE  (3)    = -.83823E-02
F21AP  (1)    = 0.          F21AP  (2)    = 0.          F21AP  (3)    = 0.          T21AP  (1)    = 0.
T21AP  (2)    = 0.          T21AP  (3)    = 0.          F21DR  (1)    = 0.          F21DR  (2)    = 0.
F21DR  (3)    = 0.          T21DR  (1)    = 0.          T21DR  (2)    = 0.          T21DR  (3)    = 0.
ALPAS         = 0.          BETAS         = 0.          VM AS         = 0.          ELECS         = 0.
S2 IT2        = -30.000

         TIME =  .1600
SRPSE  (1)    =  141.88     SRPSE  (2)    = -.48843E-03 SRPSE  (3)    = -902.81     ESTSE  (1)    = -.27008E-02
ESTSE  (2)    =  12.986     ESTSE  (3)    =  .83111E-02 XAPAE  (1)    =  128.00     XAPAE  (2)    =  .24348E-04
XAPAE  (3)    = -900.00     EAPAE  (1)    =  .81080E-08 EAPAE  (2)    =  1.2387     EAPAE  (3)    = -.87878E-02
F21AP  (1)    = 0.          F21AP  (2)    = 0.          F21AP  (3)    = 0.          T21AP  (1)    = 0.
T21AP  (2)    = 0.          T21AP  (3)    = 0.          F21DR  (1)    = 0.          F21DR  (2)    = 0.
F21DR  (3)    = 0.          T21DR  (1)    = 0.          T21DR  (2)    = 0.          T21DR  (3)    = 0.
ALPAS         = 0.          BETAS         = 0.          VM AS         = 0.          ELECS         = 0.
S2 IT2        = -30.000

         TIME =  .1800
SRPSE  (1)    =  157.40     SRPSE  (2)    = -.71003E-03 SRPSE  (3)    = -903.30     ESTSE  (1)    = -.11798E-02
ESTSE  (2)    =  13.261     ESTSE  (3)    = -.48808E-02 XAPAE  (1)    =  144.01     XAPAE  (2)    =  .27987E-04
XAPAE  (3)    = -900.00     EAPAE  (1)    =  .22841E-04 EAPAE  (2)    =  1.2078     EAPAE  (3)    = -.13723E-01
F21AP  (1)    = 0.          F21AP  (2)    = 0.          F21AP  (3)    = 0.          T21AP  (1)    = 0.
T21AP  (2)    = 0.          T21AP  (3)    = 0.          F21DR  (1)    = 0.          F21DR  (2)    = 0.
F21DR  (3)    = 0.          T21DR  (1)    = 0.          T21DR  (2)    = 0.          T21DR  (3)    = 0.
ALPAS         =  11.337     BETAS         = -.83989E-03 VM AS         =  .71114     ELECS         = 0
S2 IT2        = -30.000

         TIME =  .2000
SRPSE  (1)    =  173.22     SRPSE  (2)    = -.10878E-02 SRPSE  (3)    = -903.87     ESTSE  (1)    = -.30822E-02
ESTSE  (2)    =  13.318     ESTSE  (3)    = -.92086E-03 XAPAE  (1)    =  160.01     XAPAE  (2)    =  .29838E-04
XAPAE  (3)    = -900.00     EAPAE  (1)    =  .37822E-04 EAPAE  (2)    =  1.1734     EAPAE  (3)    = -.17811E-01
F21AP  (1)    = 0.          F21AP  (2)    = 0.          F21AP  (3)    = 0.          T21AP  (1)    = 0.
T21AP  (2)    = 0.          T21AP  (3)    = 0.          F21DR  (1)    = 0.          F21DR  (2)    = 0.
F21DR  (3)    = 0.          T21DR  (1)    = 0.          T21DR  (2)    = 0.          T21DR  (3)    = 0
ALPAS         =  11.144     BETAS         = -.37147E-03 VM AS         =  .71022     ELECS         = 0.
S2 IT2        = -30.000

  CATAPULT STRIPOFF AT TIME =       .2188 SEC

  SUSTAINER ROCKET ON AT TIME =       .2188 SEC
```

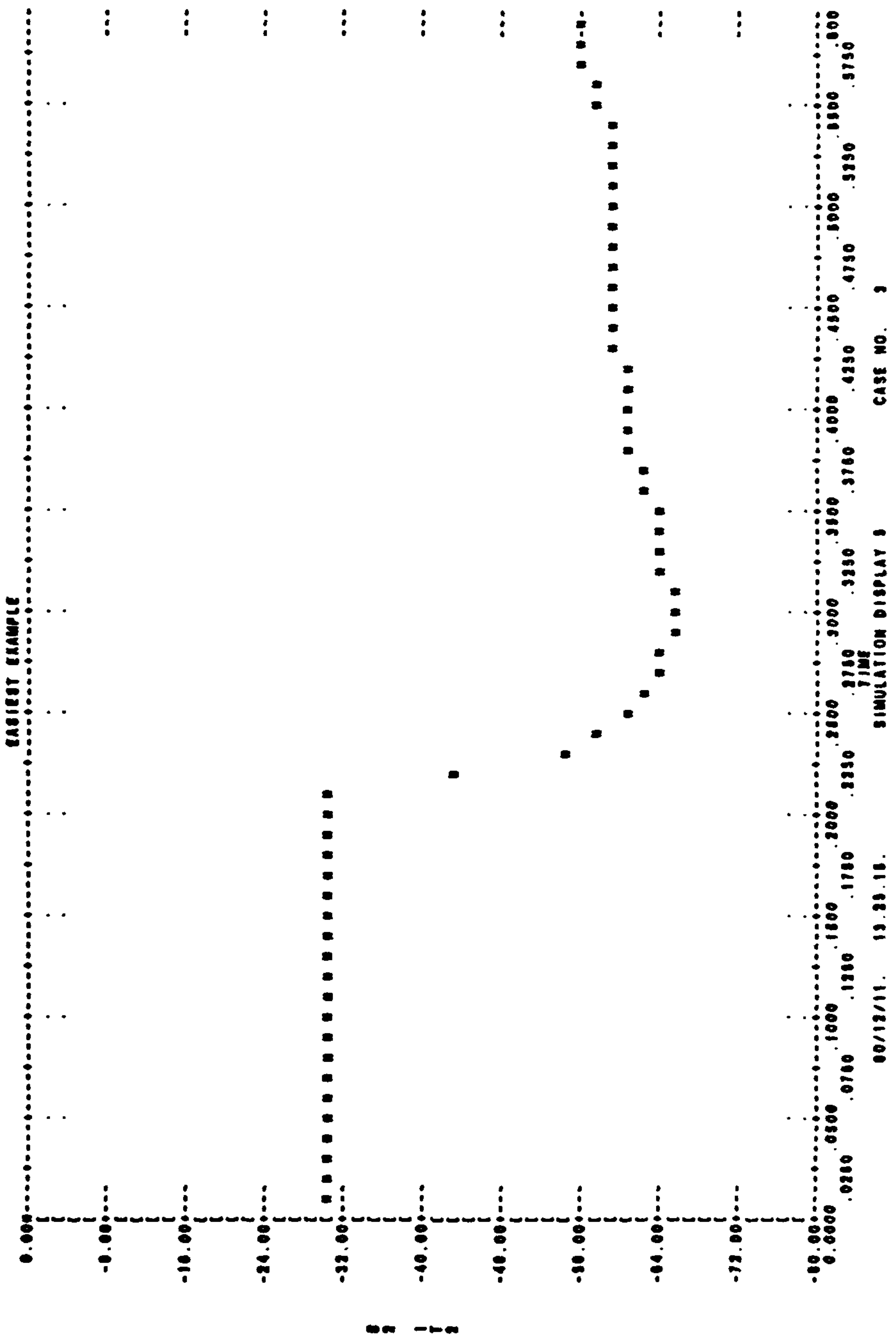
```
F21DR  (3)    = 0.          T21DR  (1)   = 0.           T21DR  (2)   = 0.          T21DR  (3)   = 0.
ALPAS         = 6.1004      BETAS        =  .03600      VM AS        =  .09573     ELECS        = -.27050
S2 IT2        = -64.637

        TIME =  .3400
SRPSE  (1)    =  282.59     SRPSE  (2)   =  .30121E-02  SRPSE  (3)   = -900.02     ESTSE  (1)   = -1.4878
ESTSE  (2)    =  7.2160     ESTSE  (3)   =  .19949      XAPAE  (1)   =  272.02     XAPAE  (2)   = -.13152E-03
XAPAE  (3)    = -899.08     EAPAE  (1)   = -.23600E-03  EAPAE  (2)   =  .04283     EAPAE  (3)   = -.39464E-01
F21AP  (1)    = -576.10     F21AP  (2)   = 0.           F21AP  (3)   =  600.52     T21AP  (1)   = 0.
T21AP  (2)    =  2001.2     T21AP  (3)   = 0.           F21DR  (1)   = 0.          F21DR  (2)   = 0.
F21DR  (3)    = 0.          T21DR  (1)   = 0.           T21DR  (2)   = 0.          T21DR  (3)   = 0.
ALPAS         =  4.8787     BETAS        =  1.8008      VM AS        =  .09307     ELECS        = -.59041
S2 IT2        = -63.753

    DART ON AT TIME =      .3575 SEC

        TIME =  .3600
SRPSE  (1)    =  297.07     SRPSE  (2)   =  .23097E-02  SRPSE  (3)   = -909.72     ESTSE  (1)   = -3.2814
ESTSE  (2)    =  7.0467     ESTSE  (3)   =  .37037      XAPAE  (1)   =  288.03     XAPAE  (2)   = -.17816E-03
XAPAE  (3)    = -899.86     EAPAE  (1)   = -.30132E-03  EAPAE  (2)   =  .03016     EAPAE  (3)   = -.37544E-01
F21AP  (1)    = -571.70     F21AP  (2)   = 0.           F21AP  (3)   =  664.24     T21AP  (1)   = 0.
T21AP  (2)    =  1888.1     T21AP  (3)   = 0.           F21DR  (1)   =  481.04     F21DR  (2)   =  94.384
F21DR  (3)    =  983.37     T21DR  (1)   = -87.708      T21DR  (2)   =  1146.3     T21DR  (3)   =  3.7833
ALPAS         =  4.4022     BETAS        =  2.3884      VM AS        =  .09030     ELECS        = -.81894
S2 IT2        = -62.632

        TIME =  .3800
SRPSE  (1)    =  313.30     SRPSE  (2)   = -.31540E-02  SRPSE  (3)   = -910.38     ESTSE  (1)   = -3.3880
ESTSE  (2)    =  7.3171     ESTSE  (3)   =  .68130      XAPAE  (1)   =  304.03     XAPAE  (2)   = -.13180E-03
XAPAE  (3)    = -890.03     EAPAE  (1)   = -.38310E-03  EAPAE  (2)   =  .03166     EAPAE  (3)   = -.36477E-01
F21AP  (1)    = -540.38     F21AP  (2)   = 0.           F21AP  (3)   =  630.20     T21AP  (1)   = 0.
T21AP  (2)    =  1686.0     T21AP  (3)   = 0.           F21DR  (1)   =  806.09     F21DR  (2)   =  54.161
F21DR  (3)    =  876.66     T21DR  (1)   = -130.18      T21DR  (2)   =  1190.8     T21DR  (3)   =  5.0606
ALPAS         =  4.0142     BETAS        =  3.2901      VM AS        =  .08740     ELECS        = -1.0782
S2 IT2        = -61.828

        TIME =  .4000
SRPSE  (1)    =  328.85     SRPSE  (2)   = -.11034E-01  SRPSE  (3)   = -911.03     ESTSE  (1)   = -4.8123
ESTSE  (2)    =  7.8004     ESTSE  (3)   =  1.1699      XAPAE  (1)   =  320.03     XAPAE  (2)   = -.20305E-03
XAPAE  (3)    = -899.80     EAPAE  (1)   = -.48894E-03  EAPAE  (2)   =  .09053     EAPAE  (3)   = -.30841E-01
F21AP  (1)    = -509.33     F21AP  (2)   = 0.           F21AP  (3)   =  602.84     T21AP  (1)   = 0.
T21AP  (2)    =  1770.0     T21AP  (3)   = 0.           F21DR  (1)   =  614.26     F21DR  (2)   =  82.619
F21DR  (3)    =  800.80     T21DR  (1)   = -210.42      T21DR  (2)   =  1204.8     T21DR  (3)   =  9.0771
ALPAS         =  5.3027     BETAS        =  4.0783      VM AS        =  .08440     ELECS        = -1.3343
S2 IT2        = -60.820

        TIME =  .4200
SRPSE  (1)    =  343.74     SRPSE  (2)   = -.26681E-01  SRPSE  (3)   = -911.88     ESTSE  (1)   = -6.3819
ESTSE  (2)    =  8.8009     ESTSE  (3)   =  1.8006      XAPAE  (1)   =  336.03     XAPAE  (2)   = -.28307E-03
XAPAE  (3)    = -899.77     EAPAE  (1)   = -.63270E-03  EAPAE  (2)   =  .08950     EAPAE  (3)   = -.20440E-01
F21AP  (1)    = -470.50     F21AP  (2)   = 0.           F21AP  (3)   =  464.41     T21AP  (1)   = 0.
T21AP  (2)    =  1634.7     T21AP  (3)   = 0.           F21DR  (1)   =  521.30     F21DR  (2)   =  122.10
F21DR  (3)    =  860.86     T21DR  (1)   = -311.28      T21DR  (2)   =  1223.3     T21DR  (3)   =  13.431
ALPAS         =  8.0287     BETAS        =  6.4028      VM AS        =  .81113     ELECS        = -1.5053
S2 IT2        = -60.013

        TIME =  .4400
SRPSE  (1)    =  380.87     SRPSE  (2)   = -.550298E-01 SRPSE  (3)   = -913.28     ESTSE  (1)   = -8.5430
ESTSE  (2)    =  9.3731     ESTSE  (3)   =  3.0411      XAPAE  (1)   =  352.04     XAPAE  (2)   = -.44277E-03
XAPAE  (3)    = -890.74     EAPAE  (1)   = -.707178E-03 EAPAE  (2)   =  1.0609     EAPAE  (3)   = -.91301E-02
F21AP  (1)    = -437.70     F21AP  (2)   = 0.           F21AP  (3)   =  432.04     T21AP  (1)   = 0.
T21AP  (2)    =  1530.8     T21AP  (3)   = 0.           F21DR  (1)   =  520.47     F21DR  (2)   =  170.88
F21DR  (3)    =  940.56     T21DR  (1)   = -460.20      T21DR  (2)   =  1238.0     T21DR  (3)   =  19.420
ALPAS         =  6.5818     BETAS        =  8.7533      VM AS        =  .67617     ELECS        = -1.8313
```

```
S2 IT2             = -60.000

             TIME =  .4000
SRPSE  (1)         =  373.02    SRPSE  (2)  = -.10530        SRPSE  (3)  = -912.00    ESTSE  (1)  = -11.184
ESTSE  (2)         =  10.100    ESTSE  (3)  =  4.7720        XAPAE  (1)  =  389.04    XAPAE  (2)  = -.53354E-03
XAPAE  (3)         = -900.70    EAPAE  (1)  = -.10201E-02    EAPAE  (2)  =  1.1307    EAPAE  (3)  =  .81052E-02
F21AP  (1)         = -410.30    F21AP  (2)  = 0.             F21AP  (3)  =  412.80    T21AP  (1)  = 0.
T21AP  (2)         =  1453.5    T21AP  (3)  = 0.             F21DR  (1)  =  820.01    F21DR  (2)  =  240.87
F21DR  (3)         =  821.81    T21DR  (1)  = -837.10        T21DR  (2)  =  1249.6    T21DR  (3)  =  27.400
ALPAS              =  0.0015    BETAS       =  11.811        VM AS       =  .07404    ELECS       = -2.0725
S2 IT2             = -50.400

             TIME =  .4000
SRPSE  (1)         =  390.09    SRPSE  (2)  = -.18010        SRPSE  (3)  = -913.40    ESTSE  (1)  = -14.100
ESTSE  (2)         =  11.030    ESTSE  (3)  =  7.3084        XAPAE  (1)  =  394.04    XAPAE  (2)  = -.63720E-03
XAPAE  (3)         = -900.80    EAPAE  (1)  = -.13040E-02    EAPAE  (2)  =  1.2403    EAPAE  (3)  =  .22812E-01
F21AP  (1)         = -410.92    F21AP  (2)  = 0.             F21AP  (3)  =  414.40    T21AP  (1)  = 0.
T21AP  (2)         =  1460.7    T21AP  (3)  = 0.             F21DR  (1)  =  820.43    F21DR  (2)  =  345.31
F21DR  (3)         =  802.02    T21DR  (1)  = -880.34        T21DR  (2)  =  1243.2    T21DR  (3)  =  87.C04
ALPAS              =  0.0227    BETAS       =  14.740        VM AS       =  .07147    ELECS       = -2.3000
S2 IT2             = -50.400

             TIME =  .5000
SRPSE  (1)         =  403.70    SRPSE  (2)  = -.31425        SRPSE  (3)  = -914.07    ESTSE  (1)  = -17.488
ESTSE  (2)         =  12.061    ESTSE  (3)  =  11.005        XAPAE  (1)  =  400.04    XAPAE  (2)  = -.78008E-03
XAPAE  (3)         = -900.02    EAPAE  (1)  = -.10074E-02    EAPAE  (2)  =  1.3881    EAPAE  (3)  =  .43120E-01
F21AP  (1)         = -450.53    F21AP  (2)  = 0.             F21AP  (3)  =  444.00    T21AP  (1)  = 0.
T21AP  (2)         =  1505.0    T21AP  (3)  = 0.             F21DR  (1)  =  814.78    F21DR  (2)  =  404.27
F21DR  (3)         =  854.08    T21DR  (1)  = -1103.0        T21DR  (2)  =  1210.7    T21DR  (3)  =  51.070
ALPAS              =  0.1020    BETAS       =  10.421        VM AS       =  .00780    ELECS       = -2.5400
S2 IT2             = -59.200

    DART OFF AT TIME =     .5005 SEC

             TIME =  .5200
SRPSE  (1)         =  410.59    SRPSE  (2)  = -.49312        SRPSE  (3)  = -914.80    ESTSE  (1)  = -21.070
ESTSE  (2)         =  13.433    ESTSE  (3)  =  16.275        XAPAE  (1)  =  416.04    XAPAE  (2)  = - .88891E-03
XAPAE  (3)         = -800.58    EAPAE  (1)  = -.30929E-02    EAPAE  (2)  =  1.5663    EAPAE  (3)  =  .60045E-01
F21AP  (1)         = -521.08    F21AP  (2)  = 0.             F21AP  (3)  =  514.23    T21AP  (1)  = 0
T21AP  (2)         =  1810.1    T21AP  (3)  = 0.             F21DR  (1)  = 0.         F21DR  (2)  = 0.
F21DR  (3)         = 0.         T21DR  (1)  = 0.             T21DR  (2)  = 0.         T21DR  (3)  = 0.
ALPAS              =  4.0883    BETAS       =  22.800        VM AS       =  .00244    ELECS       = -2.7077
S2 IT2             = -50.039

             TIME =  .5400
SRPSE  (1)         =  423.20    SRPSE  (2)  = -.72707        SRPSE  (3)  = -915.22    ESTSE  (1)  = -24.070
ESTSE  (2)         =  15.276    ESTSE  (3)  =  23.256        XAPAE  (1)  =  432.04    XAPAE  (2)  = - .10113E-02
XAPAE  (3)         = -890.84    EAPAE  (1)  = -.24757E-02    EAPAE  (2)  =  1.7531    EAPAE  (3)  =  .92037E-01
F21AP  (1)         = -850.32    F21AP  (2)  = 0.             F21AP  (3)  =  647.80    T21AP  (1)  = 0.
T21AP  (2)         =  2270.9    T21AP  (3)  = 0.             F21DR  (1)  = 0.         F21DR  (2)  = 0.
F21DR  (3)         = 0.         T21DR  (1)  = 0.             T21DR  (2)  = 0.         T21DR  (3)  = 0.
ALPAS              =  1.8402    BETAS       =  28.878        VM AS       =  .05544    ELECS       = -2.9903
S2 IT2             = -50.700

             TIME =  .5000
SRPSE  (1)         =  447.76    SRPSE  (2)  = -1.0184        SRPSE  (3)  = -915.78    ESTSE  (1)  = -28.530
ESTSE  (2)         =  18.082    ESTSE  (3)  =  32.328        XAPAE  (1)  =  446.05    XAPAE  (2)  = - .11222E-02
XAPAE  (3)         = -890.60    EAPAE  (1)  = -.27005E-02    EAPAE  (2)  =  1.9793    EAPAE  (3)  =  .12193
F21AP  (1)         = -883.73    F21AP  (2)  = 0.             F21AP  (3)  =  853.30    T21AP  (1)  = 0
T21AP  (2)         =  3000.4    T21AP  (3)  = 0.             F21DR  (1)  = 0.         F21DR  (2)  = 0.
F21DR  (3)         = 0.         T21DR  (1)  = 0.             T21DR  (2)  = 0.         T21DR  (3)  = 0.
ALPAS              = -2.7214    BETAS       =  31.074        VM AS       =  .04701    ELECS       = -3.2080
S2 IT2             = -57.227
```

```
SUSTAINER ROCKET OFF AT TIME=    .8880  SEC

          TIME =  .8800
SRPSE   (1)    =  462.07     SRPSE  (2)   = -1.3821        SRPSE  (3)   = -816.25    ESTSE  (1)   = -31.121
ESTSE   (2)    =  22.794     ESTSE  (3)   =  43.870        XAPAE  (1)   =  464.03    XAPAE  (2)   = -.12115E-02
XAPAE   (3)    = -880.48     EAPAE  (1)   = -.30164E-02    EAPAE  (2)   =  2.2230    EAPAE  (3)   =  .15334
F21AP   (1)    = -1107.8     F21AP  (2)   =  0.            F21AP  (3)   =  1083.2    T21AP  (1)   =  0.
T21AP   (2)    =  3840.2     T21AP  (3)   =  0.            F21DR  (1)   =  0.        F21DR  (2)   =  0.
F21DR   (3)    =  0.         T21DR  (1)   =  0.            T21DR  (2)   =  0.        T21DR  (3)   =  0.
ALPAS          = -6.4171     BETAS        =  34.803        VM AS        =  .84044    ELECS        = -3.4224
S2 IT2         = -30.417

                          INTEGRATOR STEP SIZE LIMITING COUNTS

                                                                        4 WB CT   = 0.        5 W1 3R   = 0.
  1 EF CT      = 0.        3 EL CT      = 0.        5 WK CT      = 0.
  6 AILCS      = 0.        7 ELECS      = 0.        8 RUDCS      = 0.
UAPAE  ( 1)    = 0.      UAPAE  ( 2)    = 0.      UAPAE  ( 3)    = 0.
XAPAE  ( 1)    = 0.      XAPAE  ( 2)    = 0.      XAPAE  ( 3)    = 0.
WAPAE  ( 1)    = 0.      WAPAE  ( 2)    = 0.      WAPAE  ( 3)    = 0.
EAPAE  ( 1)    = 0.      EAPAE  ( 2)    = 0.      EAPAE  ( 3)    = 0.      TRMAE  ( 4)   = 0.
TRMAE  ( 1)    = 0.      TRMAE  ( 2)    = 0.      TRMAE  ( 3)    = 0.
USTSE  ( 1)    = 0.      USTSE  ( 2)    = 0.      USTSE  ( 3)    = 0.
SRPSE  ( 1)    = 0.      SRPSE  ( 2)    = 0.      SRPSE  ( 3)    = 0.
WSTSE  ( 1)    = 0.      WSTSE  ( 2)    = 1.0000  WSTSE  ( 3)    = 0.
ESTSE  ( 1)    = 0.      ESTSE  ( 2)    = 0.      ESTSE  ( 3)    = 0.      40 S2 IT2   = 0.
  37 SCDSE     = 0.        38 SC SE     = 0.        39 S2 IN1    = 0.

          TIME =  .9000
SRPSE   (1)    =  470.24     SRPSE  (2)   = -1.7300        SRPSE  (3)   = -818.78    ESTSE  (1)   = -30.804
ESTSE   (2)    =  29.631     ESTSE  (3)   =  57.043        XAPAE  (1)   =  480.03    XAPAE  (2)   = -.12666E-02
XAPAE   (3)    = -899.43     EAPAE  (1)   = -.31206E-02    EAPAE  (2)   =  3.3236    EAPAE  (3)   =  .10783
F21AP   (1)    = -1341.4     F21AP  (2)   =  0.            F21AP  (3)   =  1324.7    T21AP  (1)   =  0.
T21AP   (2)    =  4083.1     T21AP  (3)   =  0.            F21DR  (1)   =  0.        F21DR  (2)   =  0.
F21DR   (3)    =  0.         T21DR  (1)   =  0.            T21DR  (2)   =  0.        T21DR  (3)   =  0.
ALPAS          = -14.383     BETAS        =  36.721        VM AS        =  .83300    ELECS        = -3.8321
S2 IT2         = -26.417

     83.8700     CPU SECONDS WERE REQUIRED FOR THE PREVIOUS ANALYSIS
```


EASIEST EXAMPLE
TIME
SIMULATION DISPLAY 2
CASE NO. 3


EASIEST EXAMPLE
30.00
27.00
24.00
21.00
18.00
15.00
12.00
9.00
6.00
3.00
0.000
0.0000
.0250
.0500
.0750
.1000
.1250
.1500
.1750
.2000
.2250
.2500
.2750
.3000
.3250
.3500
.3750
.4000
.4250
.4500
.4750
.5000
.5250
.5500
.5750
.600
TIME
80/12/11. 13.25.15. SIMULATION DISPLAY 3 CASE NO. 3



☆U.S.Government Printing Office: 1981 — 757-002/374

LMED

-8